VISUAL STUDIO .NET

EJERCICIOS

LUIS MINGUILLÓN PASCUAL

Visual Studio net

Ejercicios

Luis Minguillón Pascual

COPYRICHT
La licencia de este libro electrónico es para uso personal. Por lo tanto no puedes revenderlo a otras personas.
Gracias por respectar los derechos de autor.

Luis Minguillón Pascual

Prefacio

Desde hace tiempo he sentido una profunda convicción: el conocimiento no es algo que deba guardarse en el ámbito personal o privado, sino que debe compartirse libremente para que más personas puedan acceder a él y así ser realmente libres. Por esa razón, he decidido crear esta serie de libros de programación, con el propósito de facilitar el aprendizaje de distintos lenguajes y técnicas, ofreciendo ejercicios claros, prácticos y bien explicados.

Este libro de Visual Basic .NET es parte de ese compromiso. Aquí encontrarás una colección cuidadosamente diseñada de ejercicios que te ayudarán a desarrollar tus habilidades paso a paso, desde lo básico hasta conceptos más avanzados. Mi intención es que este material sea una herramienta útil, accesible y motivadora para quienes quieran adentrarse en el mundo de la programación o perfeccionar sus conocimientos.

Quiero aprovechar este espacio para expresar mi más sincero agradecimiento a mi esposa Marisa, cuyo apoyo incondicional ha sido fundamental en este proyecto. Su comprensión, ánimo y compañía han hecho posible que esta iniciativa vea la luz y siga creciendo.

Espero que este libro te inspire a aprender, practicar y compartir lo que descubras, porque solo así el conocimiento cumplirá su verdadero propósito: hacernos más libres.

Luis Minguillón Pascual

Indice

Índice de "200 ejercicios vb consola"

Nivel 1: Básico

- **Ejercicio 1: Mostrar un mensaje por consola**
  - Escribe un programa que muestre el mensaje: "¡Hola, mundo!" en la consola.
- **Ejercicio 2: Sumar dos números**
  - Pide al usuario que ingrese dos números y muestra la suma de ellos.
- **Ejercicio 3: Número positivo o negativo**
  - Solicita un número y muestra si es positivo, negativo o cero.
- **Ejercicio 4: Área de un triángulo**
  - Pide la base y la altura de un triángulo y calcula su área.
- **Ejercicio 5: Mostrar números del 1 al 10**
  - Usa un bucle For para mostrar en consola los números del 1 al 10.
- **Ejercicio 6: Contar números pares del 1 al 100**
  - Cuenta y muestra cuántos números pares hay entre 1 y 100.
- **Ejercicio 7: Convertir cadena a mayúsculas**
  - Pide una frase y muéstrala en mayúsculas.
- **Ejercicio 8: Función que calcula el cuadrado**
  - Crea una función que reciba un número y devuelva su cuadrado. Usa esa función en Main.
- **Ejercicio 9: Validar vocal**
  - Pide una letra y verifica si es una vocal (a, e, i, o, u).
- **Ejercicio 10: Mostrar día de la semana con Select Case**
  - Pide un número del 1 al 7 y muestra el día de la semana correspondiente.
- **Ejercicio 11: Mostrar números pares del 2 al 20**
  - Imprime en consola todos los números pares desde el 2 hasta el 20.
- **Ejercicio 12: Sumar números del 1 al 100**
  - Calcula y muestra la suma de los números del 1 al 100.
- **Ejercicio 13: Mostrar tabla de multiplicar**
  - Pide un número y muestra su tabla de multiplicar del 1 al 10.
- **Ejercicio 14: Número primo**
  - Pide un número y determina si es primo o no.
- **Ejercicio 15: Contar vocales en una frase**
  - Pide una frase y muestra cuántas vocales contiene.
- **Ejercicio 16: Número mayor entre tres**
  - Pide tres números y muestra cuál es el mayor.
- **Ejercicio 17: Convertir grados Celsius a Fahrenheit**
  - Convierte una temperatura dada en grados Celsius a Fahrenheit.
- **Ejercicio 18: Suma de números pares ingresados**
  - Pide al usuario 5 números y muestra la suma sólo de los números pares.
- **Ejercicio 19: Mostrar caracteres de una cadena uno a uno**
  - Pide una cadena y muestra cada carácter en una línea diferente.
- **Ejercicio 20: Convertir número decimal a binario**

    - Convierte un número decimal entero a su representación en binario.
- **Ejercicio 21: Invertir una cadena**
    - Pide una cadena y muestra la cadena invertida.
- **Ejercicio 22: Número factorial**
    - Pide un número y calcula su factorial.
- **Ejercicio 23: Número capicúa**
    - Pide un número y verifica si es capicúa (se lee igual de izquierda a derecha y viceversa).
- **Ejercicio 24: Convertir número romano a decimal (básico)**
    - Pide un número romano (solo I, V, X) y muestra su valor decimal.
- **Ejercicio 25: Mostrar los primeros N números de Fibonacci**
    - Pide un número N y muestra los primeros N números de la secuencia de Fibonacci.
- **Ejercicio 26: Sumar dígitos de un número**
    - Pide un número y suma todos sus dígitos.
- **Ejercicio 27: Número perfecto**
    - Pide un número y verifica si es perfecto (la suma de sus divisores propios es igual al número).
- **Ejercicio 28: Mostrar una pirámide de asteriscos**
    - Pide un número y muestra una pirámide de asteriscos con esa altura.
- **Ejercicio 29: Convertir minutos a horas y minutos**
    - Pide un número de minutos y convierte a horas y minutos (ej. 130 min = 2 horas y 10 minutos).
- **Ejercicio 30: Juego adivina el número**
    - Crea un juego en que la computadora genera un número aleatorio entre 1 y 50 y el usuario intenta adivinarlo.
- **Ejercicio 31: Calcular el área de un círculo**
    - Pide el radio y calcula el área del círculo (área = $\pi * r^2$).
- **Ejercicio 32: Contar palabras en una frase**
    - Pide una frase y cuenta cuántas palabras contiene.
- **Ejercicio 33: Mostrar los números impares entre 1 y 50**
    - Imprime en consola todos los números impares del 1 al 50.
- **Ejercicio 34: Convertir segundos a horas, minutos y segundos**
    - Pide un número de segundos y muestra la conversión a horas, minutos y segundos.
- **Ejercicio 35: Mostrar números divisibles por 3 y 5 entre 1 y 100**
    - Imprime los números entre 1 y 100 que sean divisibles por 3 y 5.
- **Ejercicio 36: Validar contraseña**
    - Pide una contraseña y verifica que tenga al menos 8 caracteres, una mayúscula y un número.
- **Ejercicio 37: Mostrar números del 10 al 1**
    - Imprime en consola los números del 10 al 1 en orden descendente.
- **Ejercicio 38: Sumar números hasta que el usuario escriba 0**
    - Pide números al usuario hasta que ingrese 0. Luego muestra la suma total.
- **Ejercicio 39: Mostrar el número más pequeño de una lista**
    - Pide al usuario 5 números y muestra el más pequeño.
- **Ejercicio 40: Mostrar números pares entre dos números dados**
    - Pide dos números y muestra todos los números pares entre ellos (inclusive).

Nivel 2: Intermedio

- **Ejercicio 41: Contar vocales en una cadena**
  - Pide una frase y cuenta cuántas vocales (a, e, i, o, u) tiene.
- **Ejercicio 42: Calcular el MCD de dos números**
  - Pide dos números y calcula su Máximo Común Divisor (MCD).
- **Ejercicio 43: Mostrar tabla de multiplicar de un número**
  - Pide un número y muestra su tabla de multiplicar del 1 al 10.
- **Ejercicio 44: Ordenar tres números de menor a mayor**
  - Pide tres números y muéstralos ordenados de menor a mayor.
- **Ejercicio 45: Validar un número primo**
  - Pide un número y verifica si es primo.
- **Ejercicio 46: Contar números positivos y negativos en una lista**
  - Pide 10 números y muestra cuántos son positivos y cuántos negativos.
- **Ejercicio 47: Contar ocurrencias de un carácter en una cadena**
  - Pide una cadena y un carácter, cuenta cuántas veces aparece ese carácter.
- **Ejercicio 48: Calcular la media de N números**
  - Pide al usuario N números y calcula la media.
- **Ejercicio 49: Generar un número aleatorio en un rango**
  - Pide dos números y genera un número aleatorio entre ellos (inclusive).
- **Ejercicio 50: Convertir temperatura Celsius a Fahrenheit y Kelvin**
  - Pide una temperatura en Celsius y muestra la conversión a Fahrenheit y Kelvin.
- **Ejercicio 51: Invertir una cadena**
  - Pide una cadena y muestra la cadena invertida.
- **Ejercicio 52: Calcular factorial de un número**
  - Pide un número y calcula su factorial.
- **Ejercicio 53: Comprobar si una cadena es palíndromo**
  - Pide una cadena y verifica si es palíndromo (se lee igual al derecho y al revés).
- **Ejercicio 54: Mostrar los primeros N números Fibonacci**
  - Pide N y muestra los primeros N números de la secuencia de Fibonacci.
- **Ejercicio 55: Calcular el número de dígitos de un número entero**
  - Pide un número entero y muestra cuántos dígitos tiene.
- **Ejercicio 56: Calcular la suma de los dígitos de un número**
  - Pide un número y suma sus dígitos.
- **Ejercicio 57: Calcular la potencia sin usar Math.Pow**
  - Pide base y exponente y calcula la potencia usando un bucle.
- **Ejercicio 58: Mostrar números pares y su suma entre 1 y N**
  - Pide N, muestra todos los números pares desde 1 a N y la suma de esos números.
- **Ejercicio 59: Calcular el área y perímetro de un rectángulo**
  - Pide base y altura, calcula y muestra el área y perímetro.
- **Ejercicio 60: Calcular el descuento aplicado a un precio**
  - Pide precio y porcentaje de descuento, calcula el precio final.
- **Ejercicio 61: Determinar si un año es bisiesto**
  - Pide un año y determina si es bisiesto o no.
- **Ejercicio 62: Convertir segundos a horas, minutos y segundos**
  - Pide un número de segundos y convierte a horas, minutos y segundos.

- **Ejercicio 63: Contar palabras en una frase**
    - Pide una frase y cuenta cuántas palabras tiene.
- **Ejercicio 64: Mostrar los números del 1 al 100 pero sustituyendo múltiplos de 3 por "Fizz", múltiplos de 5 por "Buzz"**
    - Imprime números del 1 al 100. Para múltiplos de 3 muestra "Fizz", para múltiplos de 5 "Buzz", para ambos "FizzBuzz".
- **Ejercicio 65: Solicitar contraseña y validar con 3 intentos**
    - Pide una contraseña y da hasta 3 intentos para que el usuario la ingrese correctamente.
- **Ejercicio 66: Mostrar la serie de números impares hasta N**
    - Pide N y muestra todos los números impares desde 1 hasta N.
- **Ejercicio 67: Validar si un número es capicúa (se lee igual al derecho y al revés)**
    - Pide un número y verifica si es capicúa.
- **Ejercicio 68: Convertir número decimal a binario**
    - Pide un número decimal y muestra su representación en binario.
- **Ejercicio 69: Calcular el promedio de números positivos ingresados (hasta que se ingrese cero)**
    - Pide números positivos hasta que se ingrese un 0. Muestra el promedio.
- **Ejercicio 70: Mostrar una pirámide de asteriscos con N filas**
    - Pide N y muestra una pirámide de asteriscos con N filas.
- **Ejercicio 71: Calcular el máximo común divisor (MCD) de dos números**
    - Pide dos números y calcula su MCD usando el algoritmo de Euclides.
- **Ejercicio 72: Calcular el mínimo común múltiplo (MCM) de dos números**
    - Pide dos números y calcula su MCM usando el MCD.
- **Ejercicio 73: Determinar si un número es primo**
    - Pide un número y determina si es primo.
- **Ejercicio 74: Mostrar la tabla de multiplicar de un número**
    - Pide un número y muestra su tabla de multiplicar del 1 al 10.
- **Ejercicio 75: Contar cuántas vocales tiene una frase**
    - Pide una frase y muestra cuántas vocales contiene.
- **Ejercicio 76: Ordenar tres números de menor a mayor**
    - Pide tres números y muéstralos ordenados ascendentemente.
- **Ejercicio 77: Calcular la suma de los primeros N números naturales**
    - Pide N y muestra la suma desde 1 hasta N.
- **Ejercicio 78: Mostrar los divisores de un número**
    - Pide un número y muestra todos sus divisores.
- **Ejercicio 79: Calcular la media y la desviación estándar de 5 números**
    - Pide 5 números, calcula y muestra la media y desviación estándar.
- **Ejercicio 80: Convertir grados Celsius a Fahrenheit y Kelvin**
    - Pide grados Celsius y muestra su equivalente en Fahrenheit y Kelvin.
- **Ejercicio 81: Mostrar los primeros N números Fibonacci**
    - Pide N para la serie Fibonacci y los muestra.
- **Ejercicio 82: Contar dígitos de un número**
    - Pide un número y cuenta sus dígitos.
- **Ejercicio 83: Invertir una cadena**
    - Pide una cadena y muestra la cadena invertida.

- **Ejercicio 84: Generar N números aleatorios entre 1 y 100**
    - Pide la cantidad N de números y genera y muestra N números aleatorios entre 1 y 100.
- **Ejercicio 85: Calcular el factorial de un número**
    - Pide un número y calcula su factorial.
- **Ejercicio 86: Comprobar si una palabra es palíndromo**
    - Pide una palabra o frase y verifica si es palíndromo.
- **Ejercicio 87: Calcular la suma de dígitos de un número**
    - Pide un número y suma sus dígitos.
- **Ejercicio 88: Contar cuántas veces aparece una letra en una frase**
    - Pide una frase y una letra, luego cuenta cuántas veces aparece esa letra.
- **Ejercicio 89: Calcular el promedio de números introducidos hasta que el usuario escriba -1**
    - Pide números al usuario hasta que ingrese -1 y luego muestra el promedio.
- **Ejercicio 90: Mostrar un menú con opciones y ejecutar según elección**
    - Muestra un menú con opciones (Saludar, Mostrar fecha actual, Salir) y ejecuta la opción elegida.
- **Ejercicio 91: Contar cuántos números pares hay en una lista**
    - Pide una cantidad N de números y cuenta cuántos de ellos son pares.
- **Ejercicio 92: Mostrar el número mayor y menor de una lista**
    - Pide N números y muestra el mayor y el menor de ellos.
- **Ejercicio 93: Ingresar palabras hasta que se escriba "fin" y mostrarlas ordenadas**
    - Permite ingresar palabras hasta que se escribe "fin" y luego las muestra ordenadas alfabéticamente.
- **Ejercicio 94: Mostrar un triángulo de asteriscos con N filas**
    - Pide N y muestra un triángulo de asteriscos con esa altura.
- **Ejercicio 95: Verificar si un número es capicúa (palíndromo numérico)**
    - Pide un número y verifica si es capicúa.
- **Ejercicio 96: Mostrar los números primos entre 1 y N**
    - Pide N y muestra todos los números primos desde 1 hasta N.
- **Ejercicio 97: Calcular el área de un círculo dado su radio**
    - Pide el radio de un círculo y calcula su área.
- **Ejercicio 98: Convertir un número decimal a binario**
    - Pide un número decimal y lo convierte a binario.
- **Ejercicio 99: Mostrar los primeros N números pares**
    - Pide N y muestra los primeros N números pares.
- **Ejercicio 100: Mostrar una pirámide numérica con N filas**
    - Pide el número de filas N y muestra una pirámide numérica.
- **Ejercicio 101: Calcular la suma de números pares entre 1 y N**
    - Pide N y calcula la suma de todos los números pares entre 1 y N.
- **Ejercicio 102: Determinar si un año es bisiesto**
    - Pide un año y determina si es bisiesto.
- **Ejercicio 103: Calcular la media y la desviación estándar de N números**
    - Pide N números, y calcula su media y desviación estándar.
- **Ejercicio 104: Generar y mostrar una matriz N x M con números aleatorios entre 1 y 50**

    - Pide las dimensiones N y M, y genera y muestra una matriz con números aleatorios.
- **Ejercicio 105: Transponer una matriz cuadrada N x N**
    - Pide el tamaño N de una matriz cuadrada, genera una matriz, y muestra su transpuesta.
- **Ejercicio 106: Contar vocales y consonantes en una cadena**
    - Pide una cadena y cuenta cuántas vocales y consonantes contiene.
- **Ejercicio 107: Convertir grados Celsius a Fahrenheit y Kelvin**
    - Pide una temperatura en Celsius y la convierte a Fahrenheit y Kelvin.
- **Ejercicio 108: Generar una lista de N números y ordenarla descendentemente**
    - Pide una cantidad N de números, los ingresa en una lista y la ordena descendentemente.
- **Ejercicio 109: Mostrar la tabla de multiplicar de un número**
    - Pide un número y muestra su tabla de multiplicar del 1 al 10.
- **Ejercicio 110: Detectar si una cadena contiene sólo números**
    - Pide una cadena y verifica si contiene solo números.
- **Ejercicio 111: Calcular el factorial de un número usando recursión**
    - Pide un número y calcula su factorial usando una función recursiva.
- **Ejercicio 112: Invertir una cadena de texto**
    - Pide una cadena de texto y la muestra invertida.
- **Ejercicio 113: Calcular la suma de dígitos de un número**
    - Pide un número y calcula la suma de sus dígitos.
- **Ejercicio 114: Validar si un email es correcto (básico)**
    - Pide un email y realiza una validación básica.
- **Ejercicio 115: Mostrar números Fibonacci hasta un límite N**
    - Pide un límite N y muestra los números de Fibonacci hasta ese límite.
- **Ejercicio 116: Calcular la media de números pares ingresados hasta que se ingrese un número impar**
    - Pide números hasta que se ingresa un impar, y calcula la media solo de los pares.
- **Ejercicio 117: Contar cuántas palabras tiene una frase**
    - Pide una frase y cuenta la cantidad de palabras.
- **Ejercicio 118: Calcular el mínimo común múltiplo (MCM) de dos números**
    - Pide dos números y calcula su MCM.
- **Ejercicio 119: Generar N números aleatorios únicos entre 1 y 100**
    - Pide N y genera N números aleatorios únicos entre 1 y 100.
- **Ejercicio 120: Mostrar los divisores de un número**
    - Pide un número y muestra todos sus divisores.
- **Ejercicio 121: Convertir número decimal a binario**
    - Pide un número decimal y lo convierte a binario.
- **Ejercicio 122: Calcular la potencia de un número (base y exponente) sin usar Math.Pow**
    - Pide la base y el exponente y calcula la potencia sin usar la función Math.Pow.
- **Ejercicio 123: Encontrar el elemento máximo y mínimo en una lista**
    - Pide una cantidad de números y encuentra el máximo y mínimo de la lista.
- **Ejercicio 124: Calcular la suma de una serie aritmética (N términos)**
    - Pide el primer término, la diferencia común y el número de términos para calcular la suma de una serie aritmética.

- **Ejercicio 125: Contar letras mayúsculas y minúsculas en una cadena**
  - Pide una cadena y cuenta la cantidad de letras mayúsculas y minúsculas.
- **Ejercicio 126: Comprobar si un número es primo**
  - Pide un número y verifica si es primo.
- **Ejercicio 127: Mostrar una pirámide de asteriscos**
  - Pide la altura y muestra una pirámide de asteriscos.
- **Ejercicio 128: Ordenar un arreglo usando el método de burbuja**
  - Pide un arreglo de números y lo ordena usando el método de burbuja.
- **Ejercicio 129: Convertir número binario a decimal**
  - Pide un número binario y lo convierte a decimal.
- **Ejercicio 130: Calcular el área y perímetro de un rectángulo**
  - Pide la base y altura de un rectángulo y calcula su área y perímetro.
- **Ejercicio 131: Calcular el número de dígitos de un número entero**
  - Pide un número entero y cuenta la cantidad de dígitos que tiene.
- **Ejercicio 132: Reemplazar todas las vocales en una cadena por un carácter dado**
  - Pide una cadena y un carácter, y reemplaza todas las vocales por el carácter dado.
- **Ejercicio 133: Mostrar los primeros N números primos**
  - Pide N y muestra los primeros N números primos.
- **Ejercicio 134: Contar caracteres alfabéticos, dígitos y otros en una cadena**
  - Pide una cadena y cuenta cuántos caracteres son letras, cuántos dígitos y cuántos otros.
- **Ejercicio 135: Generar una matriz identidad de tamaño N x N**
  - Pide el tamaño N y genera y muestra una matriz identidad N x N.
- **Ejercicio 136: Calcular la suma de números pares en una lista**
  - Pide una cantidad de números, los ingresa en una lista y suma solo los pares.
- **Ejercicio 137: Determinar si una cadena es un palíndromo (ignorar mayúsculas y espacios)**
  - Pide una cadena y verifica si es un palíndromo, ignorando mayúsculas y espacios.
- **Ejercicio 138: Encontrar el segundo número más grande en un arreglo**
  - Pide un arreglo de números y encuentra el segundo número más grande.
- **Ejercicio 139: Mostrar tabla de multiplicar de un número hasta 12**
  - Pide un número y muestra su tabla de multiplicar del 1 al 12.
- **Ejercicio 140: Eliminar espacios extras en una cadena (dejar solo uno entre palabras)**
  - Pide una cadena y elimina los espacios extra, dejando solo uno entre palabras.
- **Ejercicio 141: Calcular el factorial de un número usando recursión**
  - Pide un número y calcula su factorial utilizando recursión.
- **Ejercicio 142: Invertir una cadena sin usar funciones incorporadas**
  - Pide una cadena y la invierte sin usar funciones incorporadas (como StrReverse).
- **Ejercicio 143: Calcular el máximo común divisor (MCD) usando algoritmo de Euclides**
  - Pide dos números y calcula su MCD usando el algoritmo de Euclides.
- **Ejercicio 144: Mostrar números Fibonacci hasta un límite dado**
  - Pide un límite y muestra los números de la serie Fibonacci hasta ese límite.
- **Ejercicio 145: Determinar si un año es bisiesto**
  - Pide un año y determina si es bisiesto.
- **Ejercicio 146: Contar cuántas palabras hay en una cadena**
  - Pide una cadena y cuenta la cantidad de palabras que contiene.

- **Ejercicio 147: Calcular la media aritmética de números ingresados por el usuario**
  - Pide N números y calcula su media aritmética.
- **Ejercicio 148: Buscar un elemento en un arreglo y mostrar su posición (si existe)**
  - Pide un arreglo y un número a buscar, luego muestra su posición si lo encuentra.
- **Ejercicio 149: Calcular el promedio de notas y determinar si aprueba (promedio >= 6)**
  - Pide notas, calcula el promedio y determina si el estudiante aprueba (promedio >= 6).
- **Ejercicio 150: Imprimir los números pares entre dos números dados**
  - Pide dos números y muestra todos los números pares entre ellos.
- **Ejercicio 151: Contar cuántas veces aparece un carácter en una cadena**
  - Pide una cadena y un carácter, y cuenta cuántas veces aparece ese carácter.
- **Ejercicio 152: Calcular el número de palabras que comienzan con una letra específica**
  - Pide una cadena y una letra, y cuenta cuántas palabras en la cadena comienzan con esa letra.
- **Ejercicio 153: Encontrar el número más pequeño y el más grande en un arreglo**
  - Pide un arreglo de números y encuentra el más pequeño y el más grande.
- **Ejercicio 154: Verificar si un número es perfecto (suma de divisores = número)**
  - Pide un número y verifica si es un número perfecto.
- **Ejercicio 155: Calcular el área de un triángulo dados sus tres lados (fórmula de Herón)**
  - Pide los tres lados de un triángulo y calcula su área usando la fórmula de Herón.
- **Ejercicio 156: Mostrar números múltiplos de 3 entre 1 y N**
  - Pide un límite N y muestra todos los números múltiplos de 3 entre 1 y N.
- **Ejercicio 157: Convertir una temperatura de Celsius a Fahrenheit**
  - Pide una temperatura en Celsius y la convierte a Fahrenheit.
- **Ejercicio 158: Generar una pirámide de asteriscos con N niveles**
  - Pide N niveles y genera una pirámide de asteriscos.
- **Ejercicio 159: Calcular la suma de dígitos de un número**
  - Pide un número y calcula la suma de sus dígitos.
- **Ejercicio 160: Ordenar un arreglo de números enteros de menor a mayor**
  - Pide un arreglo de números enteros y los ordena de menor a mayor.
- **Ejercicio 161: Contar cuántos números negativos hay en un arreglo**
  - Pide un arreglo de números y cuenta cuántos de ellos son negativos.
- **Ejercicio 162: Validar si una cadena es un número entero positivo**
  - Pide una cadena y verifica si es un número entero positivo.
- **Ejercicio 163: Calcular el número de días entre dos fechas**
  - Pide dos fechas y calcula el número de días entre ellas.
- **Ejercicio 164: Convertir un número decimal a binario**
  - Pide un número decimal y lo convierte a binario.
- **Ejercicio 165: Mostrar los números primos entre dos números dados**
  - Pide dos números y muestra todos los números primos que se encuentran entre ellos.
- **Ejercicio 166: Calcular el promedio ponderado de notas con pesos distintos**
  - Pide notas y sus respectivos pesos, luego calcula el promedio ponderado.
- **Ejercicio 167: Reemplazar todas las vocales en una cadena por '*'**
  - Pide una cadena y reemplaza todas las vocales por el carácter '*'.
- **Ejercicio 168: Calcular la suma de los primeros N números pares**
  - Pide N y calcula la suma de los primeros N números pares.

- **Ejercicio 169: Mostrar un menú simple con opciones para sumar o restar dos números**
    - Presenta un menú para elegir entre sumar o restar dos números.
- **Ejercicio 170: Calcular el promedio de un arreglo de números decimales**
    - Pide un arreglo de números decimales y calcula su promedio.
- **Ejercicio 171: Verificar si una palabra es palíndromo (ignorar mayúsculas y espacios)**
    - Pide una palabra o frase y verifica si es un palíndromo, ignorando mayúsculas y espacios.
- **Ejercicio 172: Encontrar el segundo número más grande en un arreglo**
    - Pide un arreglo y encuentra el segundo número más grande.
- **Ejercicio 173: Contar la cantidad de dígitos en un número entero**
    - Pide un número entero y cuenta la cantidad de dígitos que tiene.
- **Ejercicio 174: Calcular la serie de Fibonacci hasta N términos**
    - Pide N y muestra los primeros N términos de la serie de Fibonacci.
- **Ejercicio 175: Invertir una cadena sin usar funciones auxiliares**
    - Pide una cadena y la invierte sin usar funciones auxiliares.
- **Ejercicio 176: Mostrar la tabla de multiplicar de un número**
    - Pide un número y muestra su tabla de multiplicar del 1 al 10.
- **Ejercicio 177: Calcular la media y la mediana de un arreglo de números**
    - Pide un arreglo de números y calcula su media y mediana.
- **Ejercicio 178: Eliminar duplicados de un arreglo de enteros**
    - Pide un arreglo de enteros y elimina los duplicados.
- **Ejercicio 179: Contar cuántas palabras tiene una frase**
    - Pide una frase y cuenta cuántas palabras contiene.
- **Ejercicio 180: Calcular el factorial de un número usando recursión**
    - Pide un número y calcula su factorial usando recursión.
- **Ejercicio 181: Determinar si un número es un número perfecto**
    - Pide un número y determina si es un número perfecto.
- **Ejercicio 182: Convertir un número decimal a hexadecimal**
    - Pide un número decimal y lo convierte a hexadecimal.
- **Ejercicio 183: Contar la cantidad de palabras que empiezan con una letra dada**
    - Pide una frase y una letra, y cuenta cuántas palabras de la frase comienzan con esa letra.
- **Ejercicio 184: Calcular la suma de los dígitos de un número entero**
    - Pide un número entero y calcula la suma de sus dígitos.
- **Ejercicio 185: Mostrar un triángulo de asteriscos de altura N**
    - Pide la altura N y muestra un triángulo de asteriscos.
- **Ejercicio 186: Validar si una contraseña cumple con ciertos criterios (mayúsculas, minúsculas, dígitos)**
    - Pide una contraseña y verifica si cumple con criterios de seguridad (longitud, mayúsculas, minúsculas, dígitos).
- **Ejercicio 187: Calcular el máximo común divisor (MCD) usando el algoritmo de Euclides**
    - Pide dos números y calcula su MCD usando el algoritmo de Euclides.
- **Ejercicio 188: Mostrar los primeros N números pares**
    - Pide N y muestra los primeros N números pares.
- **Ejercicio 189: Contar cuántos caracteres de una cadena son vocales**

    - Pide una cadena y cuenta la cantidad de vocales que contiene.
- **Ejercicio 190: Sumar dos matrices 3x3**
    - Pide los elementos de dos matrices 3x3 y muestra su suma.
- **Ejercicio 191: Calcular la potencia de un número usando recursión**
    - Pide la base y el exponente y calcula la potencia usando recursión.
- **Ejercicio 192: Contar la cantidad de caracteres únicos en una cadena**
    - Pide una cadena y cuenta la cantidad de caracteres únicos que tiene.
- **Ejercicio 193: Generar números primos hasta un límite N**
    - Pide un límite N y genera y muestra todos los números primos hasta ese límite.
- **Ejercicio 194: Mostrar los dígitos de un número en orden inverso (sin convertir a cadena)**
    - Pide un número entero y muestra sus dígitos en orden inverso sin convertirlo a cadena.
- **Ejercicio 195: Ordenar un arreglo de cadenas alfabéticamente**
    - Pide una cantidad de palabras y las ordena alfabéticamente.
- **Ejercicio 196: Encontrar el índice de la primera aparición de un carácter en una cadena**
    - Pide una cadena y un carácter, y muestra el índice de la primera aparición del carácter.
- **Ejercicio 197: Calcular el promedio de notas y determinar si aprobó (nota ≥ 6)**
    - Pide una cantidad de notas, calcula el promedio y determina si el estudiante aprobó (nota ≥ 6).
- **Ejercicio 198: Validar si un número es capicúa (igual leído al derecho y al revés)**
    - Pide un número y verifica si es capicúa.
- **Ejercicio 199: Sumar los elementos de la diagonal principal de una matriz cuadrada**
    - Pide el tamaño de una matriz cuadrada, ingresa sus elementos y suma los de la diagonal principal.
- **Ejercicio 200: Contar cuántos números negativos hay en un arreglo**
    - Pide un arreglo de números y cuenta la cantidad de números negativos.

Índice de "200 ejercicios vb form"

- **Ejercicio 1: Cambiar el texto del formulario**
    - Crea un formulario que, al hacer clic en un botón, cambie su título a "¡Bienvenido!".
- **Ejercicio 2: Mostrar un mensaje al hacer clic en un botón**
    - Agrega un botón que, al pulsarlo, muestre un mensaje que diga “Hola, usuario”.
- **Ejercicio 3: Cerrar el formulario**
    - Agrega un botón que cierre el formulario actual al hacer clic.
- **Ejercicio 4: Cambiar el color de fondo del formulario**
    - Agrega un botón que, al hacer clic, cambie el fondo del formulario a azul.
- **Ejercicio 5: Cambiar el tamaño del formulario**
    - Agrega un botón que, al hacer clic, cambie el tamaño del formulario a 800x600.
- **Ejercicio 6: Mostrar la fecha y hora actual**
    - Agrega un botón que muestre la fecha y hora actual en un Label.
- **Ejercicio 7: Habilitar y deshabilitar un botón**
    - Agrega dos botones: uno que desactive el otro y uno que lo reactive.

- **Ejercicio 8: Cambiar el texto de un Label desde un TextBox**
    - Permite al usuario escribir en un TextBox y que al pulsar un botón, el texto aparezca en un Label.
- **Ejercicio 9: Mostrar un saludo personalizado**
    - Permite que el usuario escriba su nombre en un TextBox y al hacer clic en un botón, aparezca "Hola, [nombre]".
- **Ejercicio 10: Cambiar el color de texto de un Label**
    - Agrega un botón que cambie el color del texto de un Label a rojo.
- **Ejercicio 11: Limpiar un TextBox**
    - Agrega un botón que al hacer clic limpie el contenido de un TextBox.
- **Ejercicio 12: Mostrar texto en mayúsculas**
    - Al hacer clic en un botón, muestra en un Label el texto del TextBox convertido a mayúsculas.
- **Ejercicio 13: Contar caracteres de un TextBox**
    - Al hacer clic en un botón, muestra la cantidad de caracteres escritos en el TextBox.
- **Ejercicio 14: Agregar elementos a un ListBox**
    - Permite ingresar un texto en un TextBox y al pulsar un botón, añadirlo a un ListBox.
- **Ejercicio 15: Eliminar elemento seleccionado en ListBox**
    - Agrega un botón que elimine el ítem seleccionado en el ListBox.
- **Ejercicio 16: Mostrar elemento seleccionado de ListBox en Label**
    - Cuando se selecciona un ítem del ListBox, mostrarlo en un Label.
- **Ejercicio 17: Usar un ComboBox para seleccionar colores**
    - Agrega un ComboBox con colores. Al seleccionar uno, cambia el fondo del formulario a ese color.
- **Ejercicio 18: Calcular la suma de dos números**
    - Permite al usuario ingresar dos números en dos TextBox y mostrar la suma al hacer clic.
- **Ejercicio 19: Usar un CheckBox para activar/desactivar un botón**
    - Agrega un CheckBox. Si está marcado, activa un botón; si no, lo desactiva.
- **Ejercicio 20: Mostrar contraseña oculta en TextBox**
    - Agrega un CheckBox que permita ver/ocultar el contenido de un TextBox con contraseña.
- **Ejercicio 21: Usar un ProgressBar con un temporizador**
    - Agrega un ProgressBar y un Timer. Cada vez que el timer se dispare, aumenta el valor del ProgressBar.
- **Ejercicio 22: Reiniciar ProgressBar**
    - Agrega un botón que reinicie el ProgressBar a cero.
- **Ejercicio 23: Cambiar el tamaño de fuente de un Label**
    - Agrega botones para aumentar y disminuir el tamaño del texto de un Label.
- **Ejercicio 24: Mostrar fecha con DateTimePicker**
    - Muestra la fecha seleccionada de un DateTimePicker en un Label.
- **Ejercicio 25: Confirmar antes de cerrar el formulario**
    - Antes de cerrar el formulario, preguntar al usuario si está seguro.
- **Ejercicio 26: Mostrar número de clicks en botón**
    - Cada vez que se pulse un botón, contar cuántas veces se ha hecho clic.
- **Ejercicio 27: Cambiar imagen de un PictureBox**

    - Agrega un botón que cambie la imagen mostrada en un PictureBox.
- **Ejercicio 28: Reproducir un sonido WAV**
    - Agrega un botón que reproduzca un sonido .wav.
- **Ejercicio 29: Usar RadioButtons para elegir un color**
    - Agrega tres RadioButtons para elegir entre rojo, verde o azul, y aplicar el color a un Label.
- **Ejercicio 30: Mostrar hora en tiempo real con Timer**
    - Usa un Timer para actualizar cada segundo un Label con la hora actual.
- **Ejercicio 31: Bloquear el cierre del formulario si un campo está vacío**
    - Evita que se cierre el formulario si un TextBox está vacío.
- **Ejercicio 32: Centrar el formulario al iniciar**
    - Haz que el formulario se muestre centrado en la pantalla al ejecutarse.
- **Ejercicio 33: Abrir un formulario secundario desde uno principal**
    - Agrega un botón que abra otro formulario llamado Form2.
- **Ejercicio 34: Abrir un formulario como diálogo (modal)**
    - Abre Form2 como ventana modal desde Form1.
- **Ejercicio 35: Ocultar el formulario actual**
    - Al hacer clic en un botón, oculta el formulario.
- **Ejercicio 36: Cerrar un formulario con botón**
    - Agrega un botón que cierre el formulario.
- **Ejercicio 37: Mostrar número aleatorio en un Label**
    - Genera un número aleatorio entre 1 y 100 al pulsar un botón.
- **Ejercicio 38: Mostrar mensaje con el nombre de usuario ingresado**
    - Al pulsar un botón, muestra un saludo con el nombre escrito en un TextBox.
- **Ejercicio 39: Evitar que se ingresen letras en un TextBox numérico**
    - Evita que el usuario escriba letras en un campo solo numérico.
- **Ejercicio 40: Hacer un contador con botones (+1 y -1)**
    - Crea dos botones: uno para sumar 1 al contador y otro para restar 1.
- **Ejercicio 41: Cambiar el título del formulario desde un TextBox**
    - Escribe un texto en un TextBox y cambia el título del formulario.
- **Ejercicio 42: Hacer visible un control oculto**
    - Un Label está oculto. Hazlo visible al pulsar un botón.
- **Ejercicio 43: Crear un temporizador para cuenta regresiva**
    - Usa un Timer para hacer una cuenta atrás de 10 a 0 en un Label.
- **Ejercicio 44: Seleccionar archivo con OpenFileDialog**
    - Abre un cuadro para seleccionar un archivo y muestra su ruta.
- **Ejercicio 45: Cambiar el color de fondo con ColorDialog**
    - Abre un selector de color y cambia el fondo del formulario.
- **Ejercicio 46: Agregar elementos por teclado al ListBox (Enter)**
    - Permite añadir texto al ListBox al pulsar Enter en el TextBox.
- **Ejercicio 47: Contar elementos de un ListBox**
    - Al pulsar un botón, muestra cuántos elementos hay en un ListBox.
- **Ejercicio 48: Validar que un campo esté lleno antes de continuar**
    - Si el TextBox está vacío, mostrar un error y no continuar.
- **Ejercicio 49: Hacer que un botón parpadee (cambiar color)**
    - Con un Timer, cambia el color del botón cada segundo.

- **Ejercicio 50: Cargar texto desde archivo a TextBox**
  - Abre un archivo de texto y muestra su contenido en un TextBox.
- **Ejercicio 51: Guardar texto de un TextBox en un archivo**
  - Guarda el contenido de un TextBox en un archivo de texto usando SaveFileDialog.
- **Ejercicio 52: Maximizar el formulario al iniciar**
  - Haz que el formulario se abra en modo maximizado.
- **Ejercicio 53: Desactivar un botón después de pulsarlo**
  - Desactiva un botón tras hacer clic en él.
- **Ejercicio 54: Contar caracteres de un TextBox en tiempo real**
  - Muestra en un Label el número de caracteres mientras se escribe.
- **Ejercicio 55: Mostrar un reloj en tiempo real**
  - Muestra la hora actual en un Label usando un Timer.
- **Ejercicio 56: Cambiar color de fondo del formulario al azar**
  - Al hacer clic en un botón, cambia el fondo a un color aleatorio.
- **Ejercicio 57: Mostrar mensaje de despedida al cerrar**
  - Al cerrar el formulario, muestra "Gracias por usar la aplicación".
- **Ejercicio 58: Seleccionar todos los elementos de un ListBox**
  - Agrega un botón que seleccione todos los elementos del ListBox.
- **Ejercicio 59: Mostrar el índice del elemento seleccionado en un ListBox**
  - Muestra el índice del ítem seleccionado en un Label.
- **Ejercicio 60: Validar formato de correo electrónico**
  - Verifica si el texto del TextBox tiene un formato de correo válido.
- **Ejercicio 61: Mostrar el día de la semana actual**
  - Muestra el día actual (lunes, martes, etc.) en un Label.
- **Ejercicio 62: Sumar dos números ingresados en dos TextBox**
  - Ingresa dos números y muestra su suma.
- **Ejercicio 63: Mostrar el nombre del mes actual**
  - Muestra el mes actual (enero, febrero, etc.) en un Label.
- **Ejercicio 64: Hacer un TextBox de solo lectura**
  - Haz que el TextBox1 no permita que el usuario escriba.
- **Ejercicio 65: Mostrar una barra de progreso creciente**
  - Usa un Timer para aumentar el valor de una ProgressBar.
- **Ejercicio 66: Mostrar mensaje personalizado según la hora**
  - Mostrar "Buenos días", "Buenas tardes" o "Buenas noches" según la hora.
- **Ejercicio 67: Contar cuántas veces se ha pulsado un botón**
  - Cuenta y muestra cuántas veces se ha pulsado un botón.
- **Ejercicio 68: Mostrar un mensaje solo una vez**
  - Al pulsar un botón, se muestra un mensaje solo la primera vez.
- **Ejercicio 69: Desplazar un Label a la derecha con un botón**
  - Cada vez que pulses un botón, mueve un Label 10 píxeles a la derecha.
- **Ejercicio 70: Cambiar el texto de un botón al hacer clic**
  - Haz que el texto del botón cambie a "¡Clic hecho!" después de pulsarlo.
- **Ejercicio 71: Alternar visibilidad de un Label**
  - Cada vez que pulses un botón, alterna la visibilidad de un Label.
- **Ejercicio 72: Limpiar todos los TextBox del formulario**
  - Crea un botón que borre el contenido de todos los TextBox.

- **Ejercicio 73: Deshabilitar todos los botones del formulario**
  - Agrega un botón que deshabilite todos los botones del formulario, incluido él mismo.
- **Ejercicio 74: Cambiar la fuente de un Label**
  - Al hacer clic en un botón, cambia la fuente y el tamaño del texto de un Label.
- **Ejercicio 75: Hacer parpadear un Label**
  - Con un Timer, haz que un Label parpadee (aparezca y desaparezca).
- **Ejercicio 76: Cerrar el formulario desde un botón**
  - Al pulsar un botón, cierra la aplicación.
- **Ejercicio 77: Redondear un número decimal de un TextBox**
  - Al pulsar un botón, redondea el número ingresado y lo muestra en un Label.
- **Ejercicio 78: Mostrar fecha actual con formato largo**
  - Muestra la fecha actual en formato largo (por ejemplo: "jueves, 27 de junio de 2025").
- **Ejercicio 79: Contar cuántos CheckBox están marcados**
  - Muestra cuántos CheckBox están marcados en el formulario.
- **Ejercicio 80: Mostrar nombre de usuario ingresado**
  - Pide al usuario su nombre en un TextBox y lo saluda en un Label.
- **Ejercicio 81: Crear una calculadora de resta**
  - Permite ingresar dos números y muestra su resta.
- **Ejercicio 82: Establecer título del formulario con un TextBox**
  - Permite al usuario cambiar el título del formulario escribiéndolo en un TextBox.
- **Ejercicio 83: Mostrar un mensaje si un CheckBox está marcado**
  - Al hacer clic en un botón, si un CheckBox está marcado, mostrar "Aceptado".
- **Ejercicio 84: Crear un contador regresivo de 10 a 0**
  - Con un Timer, muestra un contador regresivo en un Label.
- **Ejercicio 85: Mostrar el número más grande entre dos TextBox**
  - Compara dos números y muestra cuál es el mayor.
- **Ejercicio 86: Crear una función que devuelva el doble de un número**
  - Crea una función que reciba un número y devuelva el doble. Muestra el resultado al pulsar un botón.
- **Ejercicio 87: Mostrar si un número es par o impar**
  - Ingresa un número y muestra si es par o impar.
- **Ejercicio 88: Crear un botón que aumente el tamaño del formulario**
  - Al hacer clic, agranda el formulario en 20 píxeles de ancho y alto.
- **Ejercicio 89: Mostrar cuántos caracteres tiene el nombre ingresado**
  - Cuenta cuántos caracteres tiene el texto de un TextBox.
- **Ejercicio 90: Mostrar nombre completo usando dos TextBox**
  - Concatena nombre y apellido en un Label.
- **Ejercicio 91: Convertir texto a mayúsculas**
  - Al hacer clic en un botón, convierte el contenido de un TextBox a mayúsculas.
- **Ejercicio 92: Convertir texto a minúsculas**
  - Transforma el texto de un TextBox a minúsculas al pulsar un botón.
- **Ejercicio 93: Deshabilitar un TextBox desde un botón**
  - Desactiva un TextBox haciendo clic en un botón.
- **Ejercicio 94: Contar palabras de un TextBox**
  - Cuenta cuántas palabras hay en el TextBox.

- **Ejercicio 95: Mostrar la hora actual en un Label**
  - Muestra la hora actual en un Label usando un Timer.
- **Ejercicio 96: Cambiar el color de fondo del formulario**
  - Cambia el color de fondo del formulario al pulsar un botón.
- **Ejercicio 97: Mostrar una lista de países en un ComboBox**
  - Agrega algunos países a un ComboBox en la carga del formulario.
- **Ejercicio 98: Mostrar país seleccionado de un ComboBox**
  - Muestra el país seleccionado en un Label.
- **Ejercicio 99: Eliminar un ítem seleccionado de un ListBox**
  - Elimina el elemento seleccionado del ListBox.
- **Ejercicio 100: Calcular promedio de tres notas**
  - Muestra el promedio de tres notas ingresadas en TextBox.
- **Ejercicio 101: Agregar ítems a un ListBox desde un TextBox**
  - Al pulsar un botón, añade el contenido del TextBox a un ListBox.
- **Ejercicio 102: Mostrar número de elementos en ListBox**
  - Muestra cuántos elementos hay en un ListBox.
- **Ejercicio 103: Limpiar todos los ítems del ListBox**
  - Borra todos los elementos de un ListBox.
- **Ejercicio 104: Seleccionar todos los CheckBox del formulario**
  - Marca todos los CheckBox que estén en el formulario.
- **Ejercicio 105: Calcular área de un rectángulo**
  - Ingresa base y altura y muestra el área del rectángulo.
- **Ejercicio 106: Crear un menú con opciones de color de fondo**
  - Agrega opciones a un MenuStrip para cambiar el color de fondo del formulario.
- **Ejercicio 107: Mostrar mensaje si el campo está vacío**
  - Si el TextBox está vacío, mostrar advertencia al hacer clic.
- **Ejercicio 108: Bloquear botón tras pulsarlo una vez**
  - El botón se desactiva a sí mismo tras el primer clic.
- **Ejercicio 109: Confirmar antes de cerrar el formulario**
  - Solicita confirmación al cerrar el formulario.
- **Ejercicio 110: Abrir un archivo de texto en un TextBox**
  - Carga el contenido de un archivo .txt en un TextBox.
- **Ejercicio 111: Validar que un campo contenga solo números**
  - Muestra un mensaje si el TextBox contiene letras.
- **Ejercicio 112: Contar caracteres de un TextBox**
  - Muestra la cantidad de caracteres que tiene un TextBox.
- **Ejercicio 113: Mostrar imagen en PictureBox**
  - Carga una imagen en un PictureBox desde un archivo.
- **Ejercicio 114: Alternar visibilidad de una imagen**
  - Muestra u oculta la imagen del PictureBox.
- **Ejercicio 115: Usar ProgressBar con Timer**
  - Aumenta el valor de una ProgressBar con un Timer.
- **Ejercicio 116: Cambiar texto de un botón al hacer clic**
  - El botón cambia su texto cuando se pulsa.
- **Ejercicio 117: Calcular el factorial de un número**
  - Ingresa un número y muestra su factorial.

- **Ejercicio 118: Calcular cuántas vocales tiene un texto**
  - Cuenta la cantidad de vocales en el contenido de un TextBox.
- **Ejercicio 119: Cambiar el tamaño de fuente de un Label**
  - Aumenta el tamaño de letra de un Label.
- **Ejercicio 120: Crear reloj digital en tiempo real**
  - Muestra la hora en tiempo real en un Label usando un Timer.
- **Ejercicio 121: Aceptar solo letras en un TextBox**
  - Evita que el usuario escriba números en un TextBox.
- **Ejercicio 122: Cambiar imagen de fondo del formulario**
  - Permite cargar una imagen como fondo del formulario.
- **Ejercicio 123: Mostrar una lista desplegable con nombres**
  - Muestra una lista de nombres en un ComboBox al cargar el formulario.
- **Ejercicio 124: Cambiar color de texto en un Label**
  - Cambia el color de la letra de un Label al hacer clic.
- **Ejercicio 125: Seleccionar elemento predeterminado de ComboBox**
  - Selecciona por defecto el primer elemento del ComboBox.
- **Ejercicio 126: Agregar etiquetas dinámicamente al formulario**
  - Crea y muestra una etiqueta con texto personalizado desde código.
- **Ejercicio 127: Saludar según el nombre ingresado**
  - Al pulsar el botón, saluda con el nombre del usuario.
- **Ejercicio 128: Verificar si un número es par o impar**
  - Muestra si el número ingresado es par o impar.
- **Ejercicio 129: Desplazar un Label horizontalmente con botones**
  - Dos botones permiten mover un Label a la izquierda o derecha.
- **Ejercicio 130: Simular cambio de tema claro/oscuro**
  - Un botón alterna entre fondo claro y oscuro.
- **Ejercicio 131: Sumar los elementos de un ListBox**
  - Muestra la suma de todos los elementos numéricos en un ListBox.
- **Ejercicio 132: Mostrar el mayor de tres números**
  - Compara tres números ingresados y muestra el mayor.
- **Ejercicio 133: Alternar color de fondo con CheckBox**
  - Si se activa el CheckBox, el fondo se pone azul. Si se desactiva, blanco.
- **Ejercicio 134: Calcular porcentaje**
  - Calcula el porcentaje de un valor según lo indicado por el usuario.
- **Ejercicio 135: Mostrar números pares del 1 al 100 en ListBox**
  - Llena un ListBox con números pares del 1 al 100.
- **Ejercicio 136: Mostrar días de la semana en ComboBox**
  - Llena un ComboBox con los días de la semana.
- **Ejercicio 137: Mostrar tabla de multiplicar**
  - Muestra la tabla de multiplicar del número ingresado del 1 al 10.
- **Ejercicio 138: Mostrar fecha actual**
  - Muestra la fecha actual al hacer clic en un botón.
- **Ejercicio 139: Cambiar imagen del botón**
  - Muestra una imagen en un Button.
- **Ejercicio 140: Mostrar día de la semana actual**
  - Muestra el nombre del día actual (lunes, martes, etc.).

- **Ejercicio 141: Alinear texto del Label al centro**
  - Cambia la alineación del texto del Label al centro.
- **Ejercicio 142: Invertir texto de un TextBox**
  - Invierte el texto escrito en el TextBox.
- **Ejercicio 143: Verificar si una palabra es palíndromo**
  - Verifica si el texto ingresado es un palíndromo (igual al revés).
- **Ejercicio 144: Salir del programa**
  - Cerrar la aplicación desde un botón.
- **Ejercicio 145: Mostrar cantidad de palabras**
  - Cuenta cuántas palabras contiene un TextBox.
- **Ejercicio 146: Mostrar edad según fecha de nacimiento**
  - Usa un DateTimePicker para calcular la edad del usuario.
- **Ejercicio 147: Contador de clics**
  - Cuenta cuántas veces se ha hecho clic en un botón.
- **Ejercicio 148: Validar campos obligatorios**
  - Muestra un mensaje si un TextBox está vacío.
- **Ejercicio 149: Convertir texto a mayúsculas**
  - Convierte el texto del TextBox a mayúsculas.
- **Ejercicio 150: Agregar fecha y hora a ListBox**
  - Agrega la fecha y hora actual como nueva entrada en el ListBox.
- **Ejercicio 151: Deshabilitar botón tras primer clic**
  - Haz que el botón se desactive después de hacer clic una vez.
- **Ejercicio 152: Mostrar nombre seleccionado del ComboBox**
  - Muestra el nombre seleccionado del ComboBox en un Label.
- **Ejercicio 153: Limpiar todos los TextBox del formulario**
  - Al hacer clic en un botón, borra el contenido de todos los TextBox.
- **Ejercicio 154: Convertir texto a minúsculas**
  - Convierte el contenido del TextBox a minúsculas.
- **Ejercicio 155: Contar letras en un TextBox**
  - Muestra cuántas letras tiene el texto escrito.
- **Ejercicio 156: Activar botón solo si hay texto**
  - Activa el botón solo si el TextBox no está vacío.
- **Ejercicio 157: Mostrar el número de elementos en un ListBox**
  - Cuenta cuántos elementos tiene el ListBox.
- **Ejercicio 158: Quitar último elemento del ListBox**
  - Elimina el último ítem del ListBox.
- **Ejercicio 159: Detectar tecla Enter en TextBox**
  - Al presionar Enter, muestra el contenido del TextBox en el Label.
- **Ejercicio 160: Mostrar hora actual en tiempo real**
  - Usa un Timer para actualizar la hora en un Label cada segundo.
- **Ejercicio 161: Mostrar mensaje de bienvenida personalizado**
  - Pide el nombre en un TextBox y muestra “Bienvenido, [nombre]”.
- **Ejercicio 162: Marcar todas las CheckBox de un grupo**
  - Marca todas las CheckBox del GroupBox.
- **Ejercicio 163: Borrar elemento seleccionado del ListBox**
  - Elimina el ítem seleccionado en el ListBox.

- **Ejercicio 164: Cambiar tamaño de fuente del Label**
  - Al hacer clic en el botón, cambia el tamaño de la fuente del Label.
- **Ejercicio 165: Ordenar alfabéticamente un ListBox**
  - Ordena los elementos del ListBox alfabéticamente.
- **Ejercicio 166: Mostrar selección del RadioButton**
  - Muestra cuál RadioButton fue seleccionado.
- **Ejercicio 167: Mostrar número aleatorio entre 1 y 10**
  - Genera un número aleatorio entre 1 y 10 y lo muestra.
- **Ejercicio 168: Mostrar mensaje de despedida**
  - Muestra “Gracias por usar la app” y cierra la aplicación.
- **Ejercicio 169: Seleccionar todo el texto al enfocar TextBox**
  - Cuando el TextBox recibe foco, selecciona todo su texto.
- **Ejercicio 170: Mostrar caracteres únicos del texto**
  - Muestra todos los caracteres diferentes del TextBox.
- **Ejercicio 171: Convertir texto a código ASCII**
  - Convierte cada carácter del TextBox a su valor ASCII y muéstralo separado por comas.
- **Ejercicio 172: Aumentar valor de ProgressBar al hacer clic**
  - Cada clic en el botón debe aumentar el valor del ProgressBar en 10.
- **Ejercicio 173: Mostrar cantidad de palabras**
  - Cuenta y muestra cuántas palabras hay en un TextBox.
- **Ejercicio 174: Convertir texto a título (capitalizar)**
  - Convierte el texto del TextBox a formato Reproduce un archivo de sonido WAV al hacer clic.
- **Ejercicio 178: Invertir texto ingresado**
  - Muestra el texto del TextBox al revés.
- **Ejercicio 179: Guardar texto en archivo .txt**
  - Guarda el contenido de un TextBox en un archivo de texto.
- **Ejercicio 180: Leer archivo .txt y mostrarlo**
  - Carga el contenido de un archivo .txt en un TextBox.
- **Ejercicio 181: Dibujar rectángulo en el formulario**
  - Dibuja un rectángulo rojo al hacer clic en el botón.
- **Ejercicio 182: Cambiar color de fondo del Form**
  - Cambia el color de fondo del Form al presionar un botón.
- **Ejercicio 183: Validar si un TextBox contiene solo números**
  - Verifica si el contenido del TextBox es numérico.
- **Ejercicio 184: Desactivar todos los controles del formulario**
  - Desactiva todos los controles del formulario.
- **Ejercicio 185: Simular una barra de carga**
  - Usa Timer y ProgressBar para simular una barra de carga.
- **Ejercicio 186: Cambiar el texto del botón dinámicamente**
  - Al hacer clic, cambia el texto del botón a “Procesando…”.
- **Ejercicio 187: Mostrar fecha en formato largo**
  - Muestra la fecha actual con formato largo.
- **Ejercicio 188: Alternar visibilidad de un control**
  - Oculta o muestra el Label al hacer clic.

- **Ejercicio 189: Mostrar número redondeado**
    - Redondea el número ingresado a dos decimales.
- **Ejercicio 190: Cambiar idioma del calendario (DateTimePicker)**
    - Establece el DateTimePicker en formato de fecha española.
- **].
- **Ejercicio 193: Contar clicks rápidos**
    - El usuario debe hacer el mayor número de clicks en 5 segundos.
- **Ejercicio 194: Juego “Memoria” con botones**
    - Se muestran 4 botones con colores, el usuario debe recordar el orden y repetirlo.
- **Ejercicio 195: Juego “Snake” simple**
    - Mover una caja (PictureBox) con las flechas en un panel.
- **Ejercicio 196: Juego “Pong” básico**
    - Mover un rectángulo (PictureBox) con el ratón, y otro con teclado para simular Pong.
- **Ejercicio 197: Juego “Laberinto”** responde con una imagen.
- **Ejercicio 200: Juego “Cuenta atrás”**
    - Temporizador que cuenta hacia atrás desde 10 y muestra un mensaje al acabar.

200 Ejercicios Visual Basic net de consola

Nivel 1: Básico

Ejercicios 1 a 10

Ejercicio 1: Mostrar un mensaje por consola

Enunciado:
Escribe un programa que muestre el mensaje: `"¡Hola, mundo!"` en la consola.

Explicación:
Este es el ejercicio más básico para familiarizarte con la salida en consola.

Solución:

```
Module Module1
    Sub Main()
        Console.WriteLine("¡Hola, mundo!")
        Console.ReadLine() ' Espera a que el usuario presione Enter para cerrar
    End Sub
End Module
```

Ejercicio 2: Sumar dos números

Enunciado:
Pide al usuario que ingrese dos números y muestra la suma de ellos.

Explicación:
Este ejercicio usa entrada de datos, conversión de tipos y salida.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim num1 As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el segundo número: ")
        Dim num2 As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Integer = num1 + num2
        Console.WriteLine("La suma es: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 3: Número positivo o negativo

Enunciado:
Solicita un número y muestra si es positivo, negativo o cero.

Explicación:
Usamos estructuras condicionales simples (If).

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If num > 0 Then
            Console.WriteLine("El número es positivo.")
        ElseIf num < 0 Then
            Console.WriteLine("El número es negativo.")
        Else
            Console.WriteLine("El número es cero.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 4: Área de un triángulo

Enunciado:
Pide la base y la altura de un triángulo y calcula su área.

Explicación:
Área = (base * altura) / 2

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la base del triángulo: ")
        Dim baseTri As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese la altura del triángulo: ")
        Dim alturaTri As Double = Convert.ToDouble(Console.ReadLine())

        Dim area As Double = (baseTri * alturaTri) / 2
        Console.WriteLine("El área del triángulo es: " & area)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 5: Mostrar números del 1 al 10

Enunciado:
Usa un bucle For para mostrar en consola los números del 1 al 10.

Explicación:
El bucle For es ideal para repeticiones controladas.

Solución:

```
Module Module1
    Sub Main()
        For i As Integer = 1 To 10
            Console.WriteLine(i)
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 6: Contar números pares del 1 al 100

Enunciado:
Cuenta y muestra cuántos números pares hay entre 1 y 100.

Explicación:
Se usa un bucle con condicional para contar pares.

Solución:

```
Module Module1
    Sub Main()
        Dim contador As Integer = 0
        For i As Integer = 1 To 100
            If i Mod 2 = 0 Then
                contador += 1
            End If
        Next
        Console.WriteLine("Cantidad de números pares entre 1 y 100: " &
contador)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 7: Convertir cadena a mayúsculas

Enunciado:
Pide una frase y muéstrala en mayúsculas.

Explicación:
Uso del método ToUpper() para manipulación de cadenas.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine()
        Console.WriteLine("Frase en mayúsculas: " & frase.ToUpper())
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 8: Función que calcula el cuadrado

Enunciado:
Crea una función que reciba un número y devuelva su cuadrado. Usa esa función en Main.

Explicación:
Introducción a funciones con parámetros y valor de retorno.

Solución:

```
Module Module1
    Function Cuadrado(num As Integer) As Integer
        Return num * num
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim resultado As Integer = Cuadrado(n)
        Console.WriteLine("El cuadrado es: " & resultado)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 9: Validar vocal

Enunciado:
Pide una letra y verifica si es una vocal (a, e, i, o, u).

Explicación:
Uso de condicionales y comparación de caracteres.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una letra: ")
        Dim letra As Char = Convert.ToChar(Console.ReadLine().ToLower())

        If letra = "a"c Or letra = "e"c Or letra = "i"c Or letra = "o"c Or letra
= "u"c Then
            Console.WriteLine("Es una vocal.")
        Else
            Console.WriteLine("No es una vocal.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 10: Mostrar día de la semana con Select Case

Enunciado:
Pide un número del 1 al 7 y muestra el día de la semana correspondiente.

Explicación:
Uso de Select Case para elegir entre múltiples opciones.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número del 1 al 7: ")
        Dim dia As Integer = Convert.ToInt32(Console.ReadLine())

        Select Case dia
            Case 1
                Console.WriteLine("Lunes")
            Case 2
                Console.WriteLine("Martes")
            Case 3
                Console.WriteLine("Miércoles")
            Case 4
                Console.WriteLine("Jueves")
            Case 5
                Console.WriteLine("Viernes")
            Case 6
                Console.WriteLine("Sábado")
            Case 7
                Console.WriteLine("Domingo")
            Case Else
                Console.WriteLine("Número inválido.")
        End Select
        Console.ReadLine()
    End Sub
End Module
```

Nivel 1: Básico

Ejercicios 11 a 20

Ejercicio 11: Mostrar números pares del 2 al 20

Enunciado:
Imprime en consola todos los números pares desde el 2 hasta el 20.

Explicación:
Se usa un bucle For con paso de 2.

Solución:

```
Module Module1

    Sub Main()

        For i As Integer = 2 To 20 Step 2

            Console.WriteLine(i)

        Next

        Console.ReadLine()

    End Sub
```

```
End Module
```

Ejercicio 12: Sumar números del 1 al 100

Enunciado:
Calcula y muestra la suma de los números del 1 al 100.

Explicación:
Bucle For para acumular la suma.

Solución:

```
Module Module1
    Sub Main()
        Dim suma As Integer = 0
        For i As Integer = 1 To 100
            suma += i
        Next
        Console.WriteLine("La suma de 1 a 100 es: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 13: Mostrar tabla de multiplicar

Enunciado:
Pide un número y muestra su tabla de multiplicar del 1 al 10.

Explicación:
Bucle For para multiplicar y mostrar resultados.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número para la tabla: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To 10
            Console.WriteLine($"{num} x {i} = {num * i}")
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 14: Número primo

Enunciado:
Pide un número y determina si es primo o no.

Explicación:
Un número primo sólo es divisible por 1 y por sí mismo.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim esPrimo As Boolean = True

        If num <= 1 Then
            esPrimo = False
        Else
            For i As Integer = 2 To Math.Sqrt(num)
                If num Mod i = 0 Then
                    esPrimo = False
                    Exit For
                End If
            Next
        End If

        If esPrimo Then
            Console.WriteLine(num & " es un número primo.")
        Else
            Console.WriteLine(num & " no es un número primo.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 15: Contar vocales en una frase

Enunciado:
Pide una frase y muestra cuántas vocales contiene.

Explicación:
Recorre la cadena y cuenta las vocales.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine().ToLower()
        Dim contador As Integer = 0

        For Each c As Char In frase
            If "aeiou".Contains(c) Then
                contador += 1
            End If
        Next

        Console.WriteLine("Cantidad de vocales: " & contador)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 16: Número mayor entre tres

Enunciado:
Pide tres números y muestra cuál es el mayor.

Explicación:
Se usa condicional para comparar tres números.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim n1 As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el segundo número: ")
        Dim n2 As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el tercer número: ")
        Dim n3 As Integer = Convert.ToInt32(Console.ReadLine())

        Dim mayor As Integer = n1
```

```
        If n2 > mayor Then
            mayor = n2
        End If
        If n3 > mayor Then
            mayor = n3
        End If

        Console.WriteLine("El número mayor es: " & mayor)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 17: Convertir grados Celsius a Fahrenheit

Enunciado:
Convierte una temperatura dada en grados Celsius a Fahrenheit.

Explicación:
F = (C * 9/5) + 32

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese temperatura en Celsius: ")
        Dim celsius As Double = Convert.ToDouble(Console.ReadLine())
        Dim fahrenheit As Double = (celsius * 9 / 5) + 32
        Console.WriteLine("Temperatura en Fahrenheit: " & fahrenheit)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 18: Suma de números pares ingresados

Enunciado:
Pide al usuario 5 números y muestra la suma sólo de los números pares.

Explicación:
Condicional para verificar si el número es par antes de sumar.

Solución:

```
Module Module1
```

```
    Sub Main()
        Dim sumaPares As Integer = 0

        For i As Integer = 1 To 5
            Console.Write("Ingrese número #" & i & ": ")
            Dim num As Integer = Convert.ToInt32(Console.ReadLine())

            If num Mod 2 = 0 Then
                sumaPares += num
            End If
        Next

        Console.WriteLine("Suma de números pares: " & sumaPares)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 19: Mostrar caracteres de una cadena uno a uno

Enunciado:
Pide una cadena y muestra cada carácter en una línea diferente.

Explicación:
Usa un bucle para recorrer la cadena.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()

        For Each c As Char In texto
            Console.WriteLine(c)
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 20: Convertir número decimal a binario

Enunciado:
Convierte un número decimal entero a su representación en binario.

Explicación:
Divide sucesivamente entre 2 y guarda el residuo.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número decimal: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim binario As String = ""

        If num = 0 Then
            binario = "0"
        Else
            While num > 0
                binario = (num Mod 2).ToString() & binario
                num \= 2
            End While
        End If

        Console.WriteLine("Número en binario: " & binario)
        Console.ReadLine()
    End Sub
End Module
```

Nivel 1: Básico

Ejercicios 21 a 30

Ejercicio 21: Invertir una cadena

Enunciado:
Pide una cadena y muestra la cadena invertida.

Explicación:
Recorre la cadena desde el final hacia el principio.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim invertida As String = ""

        For i As Integer = texto.Length - 1 To 0 Step -1
            invertida &= texto(i)
        Next

        Console.WriteLine("Cadena invertida: " & invertida)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 22: Número factorial

Enunciado:
Pide un número y calcula su factorial.

Explicación:
El factorial de n es el producto de todos los enteros positivos hasta n.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim factorial As Long = 1

        For i As Integer = 1 To num
            factorial *= i
        Next

        Console.WriteLine("El factorial de " & num & " es: " & factorial)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 23: Número capicúa

Enunciado:
Pide un número y verifica si es capicúa (se lee igual de izquierda a derecha y viceversa).

Explicación:
Compara la cadena con su versión invertida.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As String = Console.ReadLine()
        Dim invertido As String = StrReverse(num)
```

```
        If num = invertido Then
            Console.WriteLine("El número es capicúa.")
        Else
            Console.WriteLine("El número no es capicúa.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 24: Convertir número romano a decimal (básico)

Enunciado:
Pide un número romano (solo I, V, X) y muestra su valor decimal.

Explicación:
Usamos condiciones simples para interpretar el número romano.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número romano (I, V, X): ")
        Dim romano As String = Console.ReadLine().ToUpper()
        Dim decimalVal As Integer = 0

        Select Case romano
            Case "I"
                decimalVal = 1
            Case "V"
                decimalVal = 5
            Case "X"
                decimalVal = 10
            Case Else
                Console.WriteLine("Número romano inválido para este ejercicio.")
                Console.ReadLine()
                Exit Sub
        End Select

        Console.WriteLine("Valor decimal: " & decimalVal)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 25: Mostrar los primeros N números de Fibonacci

Enunciado:
Pide un número N y muestra los primeros N números de la secuencia de Fibonacci.

Explicación:
La secuencia empieza con 0 y 1, y cada número es la suma de los dos anteriores.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de números de Fibonacci: ")
```

```
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        Dim a As Integer = 0
        Dim b As Integer = 1
        Dim temp As Integer

        For i As Integer = 1 To n
            Console.WriteLine(a)
            temp = a + b
            a = b
            b = temp
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 26: Sumar dígitos de un número

Enunciado:
Pide un número y suma todos sus dígitos.

Explicación:
Convertimos el número a cadena y sumamos los dígitos.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As String = Console.ReadLine()
        Dim suma As Integer = 0

        For Each c As Char In num
            suma += Convert.ToInt32(c.ToString())
        Next

        Console.WriteLine("La suma de los dígitos es: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 27: Número perfecto

Enunciado:
Pide un número y verifica si es perfecto (la suma de sus divisores propios es igual al número).

Explicación:
Suma los divisores menores al número y compara.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim sumaDivisores As Integer = 0
```

```
        For i As Integer = 1 To num - 1
            If num Mod i = 0 Then
                sumaDivisores += i
            End If
        Next

        If sumaDivisores = num Then
            Console.WriteLine(num & " es un número perfecto.")
        Else
            Console.WriteLine(num & " no es un número perfecto.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 28: Mostrar una pirámide de asteriscos

Enunciado:
Pide un número y muestra una pirámide de asteriscos con esa altura.

Explicación:
Usa bucles anidados para imprimir espacios y asteriscos.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la altura de la pirámide: ")
        Dim altura As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To altura
            Console.Write(New String(" "c, altura - i))
            Console.WriteLine(New String("*"c, 2 * i - 1))
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 29: Convertir minutos a horas y minutos

Enunciado:
Pide un número de minutos y convierte a horas y minutos (ej. 130 min = 2 horas y 10 minutos).

Explicación:
Divide y usa módulo para obtener horas y minutos.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese minutos: ")
        Dim minutos As Integer = Convert.ToInt32(Console.ReadLine())
        Dim horas As Integer = minutos \ 60
        Dim minutosRestantes As Integer = minutos Mod 60
```

```
        Console.WriteLine($"{minutos} minutos son {horas} horas y
{minutosRestantes} minutos.")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 30: Juego adivina el número

Enunciado:
Crea un juego en que la computadora genera un número aleatorio entre 1 y 50 y el usuario intenta adivinarlo.

Explicación:
Usa un bucle Do While y Random para generar el número.

Solución:

```
Module Module1
    Sub Main()
        Dim rnd As New Random()
        Dim numeroSecreto As Integer = rnd.Next(1, 51)
        Dim intento As Integer = 0

        Do
            Console.Write("Adivina el número (1-50): ")
            intento = Convert.ToInt32(Console.ReadLine())

            If intento < numeroSecreto Then
                Console.WriteLine("Demasiado bajo.")
            ElseIf intento > numeroSecreto Then
                Console.WriteLine("Demasiado alto.")
            Else
                Console.WriteLine("¡Correcto! Has adivinado el número.")
            End If
        Loop While intento <> numeroSecreto

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 31: Calcular el área de un círculo

Enunciado:
Pide el radio y calcula el área del círculo (área = π * r^2).

Explicación:
Usa la constante Math.PI para π y la fórmula del área.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el radio del círculo: ")
        Dim radio As Double = Convert.ToDouble(Console.ReadLine())
        Dim area As Double = Math.PI * radio * radio
        Console.WriteLine("El área del círculo es: " & area)
```

```
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 32: Contar palabras en una frase

Enunciado:
Pide una frase y cuenta cuántas palabras contiene.

Explicación:
Divide la cadena usando espacios y cuenta los elementos.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine()
        Dim palabras() As String = frase.Split(" "c,
StringSplitOptions.RemoveEmptyEntries)
        Console.WriteLine("Número de palabras: " & palabras.Length)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 33: Mostrar los números impares entre 1 y 50

Enunciado:
Imprime en consola todos los números impares del 1 al 50.

Explicación:
Usa un bucle y condicional para filtrar los impares.

Solución:

```
Module Module1
    Sub Main()
        For i As Integer = 1 To 50 Step 2
            Console.WriteLine(i)
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 34: Convertir segundos a horas, minutos y segundos

Enunciado:
Pide un número de segundos y muestra la conversión a horas, minutos y segundos.

Explicación:
Divide sucesivamente usando división entera y módulo.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese segundos: ")
        Dim totalSegundos As Integer = Convert.ToInt32(Console.ReadLine())

        Dim horas As Integer = totalSegundos \ 3600
        Dim minutos As Integer = (totalSegundos Mod 3600) \ 60
        Dim segundos As Integer = totalSegundos Mod 60

        Console.WriteLine($"{totalSegundos} segundos equivalen a {horas} horas,
{minutos} minutos y {segundos} segundos.")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 35: Mostrar números divisibles por 3 y 5 entre 1 y 100

Enunciado:
Imprime los números entre 1 y 100 que sean divisibles por 3 y 5.

Explicación:
Condicional con operador lógico AND.

Solución:

```
Module Module1
    Sub Main()
        For i As Integer = 1 To 100
            If i Mod 3 = 0 And i Mod 5 = 0 Then
                Console.WriteLine(i)
            End If
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 36: Validar contraseña

Enunciado:
Pide una contraseña y verifica que tenga al menos 8 caracteres, una mayúscula y un número.

Explicación:
Revisa longitud y caracteres con condicionales.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una contraseña: ")
        Dim pwd As String = Console.ReadLine()
        Dim tieneMayuscula As Boolean = pwd.Any(Function(c) Char.IsUpper(c))
        Dim tieneNumero As Boolean = pwd.Any(Function(c) Char.IsDigit(c))

        If pwd.Length >= 8 And tieneMayuscula And tieneNumero Then
            Console.WriteLine("Contraseña válida.")
        Else
```

```
            Console.WriteLine("Contraseña inválida. Debe tener al menos 8
caracteres, una mayúscula y un número.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 37: Mostrar números del 10 al 1

Enunciado:
Imprime en consola los números del 10 al 1 en orden descendente.

Explicación:
Usa bucle For con paso negativo.

Solución:

```
Module Module1
    Sub Main()
        For i As Integer = 10 To 1 Step -1
            Console.WriteLine(i)
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 38: Sumar números hasta que el usuario escriba 0

Enunciado:
Pide números al usuario hasta que ingrese 0. Luego muestra la suma total.

Explicación:
Usa bucle Do While y acumula suma.

Solución:

```
Module Module1
    Sub Main()
        Dim suma As Integer = 0
        Dim num As Integer

        Do
            Console.Write("Ingrese un número (0 para terminar): ")
            num = Convert.ToInt32(Console.ReadLine())
            suma += num
        Loop While num <> 0

        Console.WriteLine("Suma total: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 39: Mostrar el número más pequeño de una lista

Enunciado:
Pide al usuario 5 números y muestra el más pequeño.

Explicación:
Compara cada número para actualizar el mínimo.

Solución:

```
Module Module1
    Sub Main()
        Dim menor As Integer = Integer.MaxValue

        For i As Integer = 1 To 5
            Console.Write("Ingrese número #" & i & ": ")
            Dim num As Integer = Convert.ToInt32(Console.ReadLine())

            If num < menor Then
                menor = num
            End If
        Next

        Console.WriteLine("El número más pequeño es: " & menor)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 40: Mostrar números pares entre dos números dados

Enunciado:
Pide dos números y muestra todos los números pares entre ellos (inclusive).

Explicación:
Usa un bucle For entre los números y condicional para pares.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim inicio As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el segundo número: ")
        Dim fin As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = inicio To fin
            If i Mod 2 = 0 Then
                Console.WriteLine(i)
            End If
        Next

        Console.ReadLine()
    End Sub
End Module
```

Nivel 2: Intermedio

Ejercicios 41 a 50

Ejercicio 41: Contar vocales en una cadena

Enunciado:
Pide una frase y cuenta cuántas vocales (a, e, i, o, u) tiene.

Explicación:
Recorre la cadena y compara cada carácter.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim texto As String = Console.ReadLine().ToLower()
        Dim contador As Integer = 0
        Dim vocales As String = "aeiou"

        For Each c As Char In texto
            If vocales.Contains(c) Then
                contador += 1
            End If
        Next

        Console.WriteLine("Número de vocales: " & contador)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 42: Calcular el MCD de dos números

Enunciado:
Pide dos números y calcula su Máximo Común Divisor (MCD).

Explicación:
Usamos el algoritmo de Euclides.

Solución:

```
Module Module1
    Function MCD(a As Integer, b As Integer) As Integer
        While b <> 0
            Dim temp As Integer = b
            b = a Mod b
            a = temp
        End While
        Return a
    End Function

    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim num1 As Integer = Convert.ToInt32(Console.ReadLine())
```

```
        Console.Write("Ingrese el segundo número: ")
        Dim num2 As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine("El MCD es: " & MCD(num1, num2))
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 43: Mostrar tabla de multiplicar de un número

Enunciado:
Pide un número y muestra su tabla de multiplicar del 1 al 10.

Explicación:
Bucle For para multiplicar y mostrar resultados.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número para la tabla de multiplicar: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To 10
            Console.WriteLine($"{num} x {i} = {num * i}")
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 44: Ordenar tres números de menor a mayor

Enunciado:
Pide tres números y muéstralos ordenados de menor a mayor.

Explicación:
Compara y ordena usando variables temporales.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese número 1: ")
        Dim a As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese número 2: ")
        Dim b As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese número 3: ")
        Dim c As Integer = Convert.ToInt32(Console.ReadLine())

        Dim temp As Integer

        If a > b Then
            temp = a
            a = b
            b = temp
        End If
```

```
        If a > c Then
            temp = a
            a = c
            c = temp
        End If

        If b > c Then
            temp = b
            b = c
            c = temp
        End If

        Console.WriteLine("Números ordenados: " & a & ", " & b & ", " & c)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 45: Validar un número primo

Enunciado:
Pide un número y verifica si es primo.

Explicación:
Un número primo solo es divisible por 1 y por sí mismo.

Solución:

```
Module Module1
    Function EsPrimo(num As Integer) As Boolean
        If num < 2 Then Return False
        For i As Integer = 2 To Math.Sqrt(num)
            If num Mod i = 0 Then
                Return False
            End If
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim numero As Integer = Convert.ToInt32(Console.ReadLine())

        If EsPrimo(numero) Then
            Console.WriteLine(numero & " es un número primo.")
        Else
            Console.WriteLine(numero & " no es un número primo.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 46: Contar números positivos y negativos en una lista

Enunciado:
Pide 10 números y muestra cuántos son positivos y cuántos negativos.

Explicación:
Usa contadores y condicionales.

Solución:

```
Module Module1
    Sub Main()
        Dim positivos As Integer = 0
        Dim negativos As Integer = 0

        For i As Integer = 1 To 10
            Console.Write("Ingrese número #" & i & ": ")
            Dim num As Integer = Convert.ToInt32(Console.ReadLine())
            If num > 0 Then
                positivos += 1
            ElseIf num < 0 Then
                negativos += 1
            End If
        Next

        Console.WriteLine("Números positivos: " & positivos)
        Console.WriteLine("Números negativos: " & negativos)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 47: Contar ocurrencias de un carácter en una cadena

Enunciado:
Pide una cadena y un carácter, cuenta cuántas veces aparece ese carácter.

Explicación:
Recorre la cadena y compara con el carácter dado.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Console.Write("Ingrese un carácter para buscar: ")
        Dim c As Char = Console.ReadLine()(0)
        Dim contador As Integer = 0

        For Each ch As Char In texto
            If ch = c Then
                contador += 1
            End If
        Next

        Console.WriteLine($"El carácter '{c}' aparece {contador} veces.")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 48: Calcular la media de N números

Enunciado:
Pide al usuario N números y calcula la media.

Explicación:
Suma todos los números y divide por N.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("¿Cuántos números quiere ingresar? ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Double = 0

        For i As Integer = 1 To n
            Console.Write("Número #" & i & ": ")
            suma += Convert.ToDouble(Console.ReadLine())
        Next

        Dim media As Double = suma / n
        Console.WriteLine("La media es: " & media)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 49: Generar un número aleatorio en un rango

Enunciado:
Pide dos números y genera un número aleatorio entre ellos (inclusive).

Explicación:
Usa Random y el rango dado.

Solución:

```
Module Module1
    Sub Main()
        Dim rnd As New Random()
        Console.Write("Ingrese el límite inferior: ")
        Dim min As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el límite superior: ")
        Dim max As Integer = Convert.ToInt32(Console.ReadLine())

        Dim numeroAleatorio As Integer = rnd.Next(min, max + 1)
        Console.WriteLine("Número aleatorio generado: " & numeroAleatorio)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 50: Convertir temperatura Celsius a Fahrenheit y Kelvin

Enunciado:
Pide una temperatura en Celsius y muestra la conversión a Fahrenheit y Kelvin.

Explicación:
Aplica las fórmulas:

- Fahrenheit = Celsius * 9/5 + 32
- Kelvin = Celsius + 273.15

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese temperatura en Celsius: ")
        Dim celsius As Double = Convert.ToDouble(Console.ReadLine())
        Dim fahrenheit As Double = celsius * 9 / 5 + 32
        Dim kelvin As Double = celsius + 273.15

        Console.WriteLine($"{celsius}°C = {fahrenheit}°F")
        Console.WriteLine($"{celsius}°C = {kelvin} K")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 51: Invertir una cadena

Enunciado:
Pide una cadena y muestra la cadena invertida.

Explicación:
Recorre la cadena desde el final para crear una nueva.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim invertido As String = ""

        For i As Integer = texto.Length - 1 To 0 Step -1
            invertido &= texto(i)
        Next

        Console.WriteLine("Cadena invertida: " & invertido)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 52: Calcular factorial de un número

Enunciado:
Pide un número y calcula su factorial.

Explicación:
Multiplica todos los números desde 1 hasta el número dado.

Solución:

```
Module Module1
    Function Factorial(n As Integer) As Long
        Dim resultado As Long = 1
        For i As Integer = 2 To n
            resultado *= i
        Next
        Return resultado
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If num < 0 Then
            Console.WriteLine("No existe factorial para números negativos.")
        Else
            Console.WriteLine($"El factorial de {num} es {Factorial(num)}")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 53: Comprobar si una cadena es palíndromo

Enunciado:
Pide una cadena y verifica si es palíndromo (se lee igual al derecho y al revés).

Explicación:
Compara la cadena original con la invertida.

Solución:

```
Module Module1
    Function EsPalindromo(texto As String) As Boolean
        Dim limpio As String = texto.ToLower().Replace(" ", "")
        Dim invertido As String = ""

        For i As Integer = limpio.Length - 1 To 0 Step -1
            invertido &= limpio(i)
        Next

        Return limpio = invertido
    End Function

    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim cadena As String = Console.ReadLine()

        If EsPalindromo(cadena) Then
            Console.WriteLine("La cadena es un palíndromo.")
        Else
            Console.WriteLine("La cadena NO es un palíndromo.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 54: Mostrar los primeros N números Fibonacci

Enunciado:
Pide N y muestra los primeros N números de la secuencia de Fibonacci.

Explicación:
La secuencia empieza con 0 y 1, cada número es la suma de los dos anteriores.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese cuántos números Fibonacci desea: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        Dim a As Integer = 0
        Dim b As Integer = 1

        For i As Integer = 1 To n
            Console.WriteLine(a)
            Dim temp As Integer = a + b
            a = b
            b = temp
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 55: Calcular el número de dígitos de un número entero

Enunciado:
Pide un número entero y muestra cuántos dígitos tiene.

Explicación:
Convierte a cadena o usa división sucesiva.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número entero: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim digitos As Integer = num.ToString().Length

        Console.WriteLine("El número tiene " & digitos & " dígitos.")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 56: Calcular la suma de los dígitos de un número

Enunciado:
Pide un número y suma sus dígitos.

Explicación:
Convierte a cadena y suma cada dígito o usa módulo.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Integer = 0
        Dim temp As Integer = Math.Abs(num)

        While temp > 0
            suma += temp Mod 10
            temp \= 10
        End While

        Console.WriteLine("Suma de dígitos: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 57: Calcular la potencia sin usar Math.Pow

Enunciado:
Pide base y exponente y calcula la potencia usando un bucle.

Explicación:
Multiplica la base por sí misma exponente veces.

Solución:

```
Module Module1
    Function Potencia(base As Double, exponente As Integer) As Double
        Dim resultado As Double = 1
        For i As Integer = 1 To exponente
            resultado *= base
        Next
        Return resultado
    End Function

    Sub Main()
        Console.Write("Ingrese la base: ")
        Dim b As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese el exponente: ")
        Dim e As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"{b} elevado a {e} es {Potencia(b, e)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 58: Mostrar números pares y su suma entre 1 y N

Enunciado:
Pide N, muestra todos los números pares desde 1 a N y la suma de esos números.

Explicación:
Recorre con bucle y acumula suma.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Integer = 0

        For i As Integer = 2 To n Step 2
            Console.WriteLine(i)
            suma += i
        Next

        Console.WriteLine("Suma de números pares: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 59: Calcular el área y perímetro de un rectángulo

Enunciado:
Pide base y altura, calcula y muestra el área y perímetro.

Explicación:
Área = base * altura; perímetro = 2*(base + altura)

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la base: ")
        Dim base As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese la altura: ")
        Dim altura As Double = Convert.ToDouble(Console.ReadLine())

        Dim area As Double = base * altura
        Dim perimetro As Double = 2 * (base + altura)

        Console.WriteLine($"Área: {area}")
        Console.WriteLine($"Perímetro: {perimetro}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 60: Calcular el descuento aplicado a un precio

Enunciado:
Pide precio y porcentaje de descuento, calcula el precio final.

Explicación:
Precio final = precio - (precio * descuento/100)

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el precio: ")
```

```
        Dim precio As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese el porcentaje de descuento: ")
        Dim descuento As Double = Convert.ToDouble(Console.ReadLine())

        Dim precioFinal As Double = precio - (precio * descuento / 100)
        Console.WriteLine($"Precio final con descuento: {precioFinal}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 61: Determinar si un año es bisiesto

Enunciado:
Pide un año y determina si es bisiesto o no.

Explicación:
Un año es bisiesto si es divisible por 4, pero no por 100, salvo que también sea divisible por 400.

Solución:

```
Module Module1
    Function EsBisiesto(anio As Integer) As Boolean
        Return (anio Mod 4 = 0 And anio Mod 100 <> 0) Or (anio Mod 400 = 0)
    End Function

    Sub Main()
        Console.Write("Ingrese un año: ")
        Dim anio As Integer = Convert.ToInt32(Console.ReadLine())

        If EsBisiesto(anio) Then
            Console.WriteLine($"{anio} es un año bisiesto.")
        Else
            Console.WriteLine($"{anio} no es un año bisiesto.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 62: Convertir segundos a horas, minutos y segundos

Enunciado:
Pide un número de segundos y convierte a horas, minutos y segundos.

Explicación:
Usa división y módulo para obtener cada unidad.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el número de segundos: ")
        Dim totalSegundos As Integer = Convert.ToInt32(Console.ReadLine())

        Dim horas As Integer = totalSegundos \ 3600
        Dim minutos As Integer = (totalSegundos Mod 3600) \ 60
        Dim segundos As Integer = totalSegundos Mod 60
```

```
        Console.WriteLine($"{horas} horas, {minutos} minutos, {segundos}
segundos")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 63: Contar palabras en una frase

Enunciado:
Pide una frase y cuenta cuántas palabras tiene.

Explicación:
Divide la cadena en palabras usando espacio como separador.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine().Trim()

        Dim palabras() As String = frase.Split(" "c,
StringSplitOptions.RemoveEmptyEntries)
        Console.WriteLine("Número de palabras: " & palabras.Length)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 64: Mostrar los números del 1 al 100 pero sustituyendo múltiplos de 3 por "Fizz", múltiplos de 5 por "Buzz"

Enunciado:
Imprime números del 1 al 100. Para múltiplos de 3 muestra "Fizz", para múltiplos de 5 "Buzz", para ambos "FizzBuzz".

Explicación:
Condiciones if para los múltiplos.

Solución:

```
Module Module1
    Sub Main()
        For i As Integer = 1 To 100
            If i Mod 15 = 0 Then
                Console.WriteLine("FizzBuzz")
            ElseIf i Mod 3 = 0 Then
                Console.WriteLine("Fizz")
            ElseIf i Mod 5 = 0 Then
                Console.WriteLine("Buzz")
            Else
                Console.WriteLine(i)
            End If
        Next
        Console.ReadLine()
    End Sub
```

```
End Module
```

Ejercicio 65: Solicitar contraseña y validar con 3 intentos

Enunciado:
Pide una contraseña y da hasta 3 intentos para que el usuario la ingrese correctamente.

Explicación:
Usar bucle y contador de intentos.

Solución:

```
Module Module1
    Sub Main()
        Const contraseña As String = "secreto"
        Dim intentos As Integer = 0
        Dim entrada As String = ""

        While intentos < 3
            Console.Write("Ingrese la contraseña: ")
            entrada = Console.ReadLine()

            If entrada = contraseña Then
                Console.WriteLine("Contraseña correcta. Acceso concedido.")
                Exit While
            Else
                Console.WriteLine("Contraseña incorrecta.")
                intentos += 1
            End If
        End While

        If intentos = 3 Then
            Console.WriteLine("Número máximo de intentos alcanzado. Acceso
denegado.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 66: Mostrar la serie de números impares hasta N

Enunciado:
Pide N y muestra todos los números impares desde 1 hasta N.

Explicación:
Recorre con paso 2 o verifica imparidad.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n Step 2
            Console.WriteLine(i)
```

```
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 67: Validar si un número es capicúa (se lee igual al derecho y al revés)

Enunciado:
Pide un número y verifica si es capicúa.

Explicación:
Convierte a cadena y compara con su invertida.

Solución:

```
Module Module1
    Function EsCapicua(num As Integer) As Boolean
        Dim texto As String = num.ToString()
        Dim invertido As String = ""

        For i As Integer = texto.Length - 1 To 0 Step -1
            invertido &= texto(i)
        Next

        Return texto = invertido
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If EsCapicua(num) Then
            Console.WriteLine(num & " es capicúa.")
        Else
            Console.WriteLine(num & " no es capicúa.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 68: Convertir número decimal a binario

Enunciado:
Pide un número decimal y muestra su representación en binario.

Explicación:
Divide sucesivamente entre 2 y acumula restos.

Solución:

```
Module Module1
    Function DecimalABinario(num As Integer) As String
        Dim binario As String = ""
        Dim n As Integer = num
```

```
        If n = 0 Then Return "0"

        While n > 0
            binario = (n Mod 2).ToString() & binario
            n \= 2
        End While

        Return binario
    End Function

    Sub Main()
        Console.Write("Ingrese un número decimal: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"Binario: {DecimalABinario(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 69: Calcular el promedio de números positivos ingresados (hasta que se ingrese cero)

Enunciado:
Pide números positivos hasta que se ingrese un 0. Muestra el promedio.

Explicación:
Suma y cuenta números, luego divide.

Solución:

```
Module Module1
    Sub Main()
        Dim suma As Double = 0
        Dim contador As Integer = 0
        Dim num As Integer

        Do
            Console.Write("Ingrese un número positivo (0 para terminar): ")
            num = Convert.ToInt32(Console.ReadLine())
            If num > 0 Then
                suma += num
                contador += 1
            End If
        Loop While num <> 0

        If contador > 0 Then
            Console.WriteLine("Promedio: " & (suma / contador))
        Else
            Console.WriteLine("No se ingresaron números positivos.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 70: Mostrar una pirámide de asteriscos con N filas

Enunciado:
Pide N y muestra una pirámide de asteriscos con N filas.

Explicación:
Bucle anidado para espacios y asteriscos.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el número de filas: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n
            Console.Write(New String(" "c, n - i))
            Console.WriteLine(New String("*"c, 2 * i - 1))
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 71: Calcular el máximo común divisor (MCD) de dos números

Enunciado:
Pide dos números y calcula su MCD usando el algoritmo de Euclides.

Solución:

```
Module Module1
    Function MCD(a As Integer, b As Integer) As Integer
        While b <> 0
            Dim temp As Integer = b
            b = a Mod b
            a = temp
        End While
        Return a
    End Function

    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim num1 As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el segundo número: ")
        Dim num2 As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine("MCD: " & MCD(num1, num2))
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 72: Calcular el mínimo común múltiplo (MCM) de dos números

Enunciado:
Pide dos números y calcula su MCM usando el MCD.

Solución:

```
Module Module1
    Function MCD(a As Integer, b As Integer) As Integer
        While b <> 0
            Dim temp As Integer = b
            b = a Mod b
            a = temp
        End While
        Return a
    End Function

    Function MCM(a As Integer, b As Integer) As Integer
        Return (a * b) \ MCD(a, b)
    End Function

    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim num1 As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el segundo número: ")
        Dim num2 As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine("MCM: " & MCM(num1, num2))
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 73: Determinar si un número es primo

Enunciado:
Pide un número y determina si es primo.

Solución:

```
Module Module1
    Function EsPrimo(num As Integer) As Boolean
        If num < 2 Then Return False
        For i As Integer = 2 To Math.Sqrt(num)
            If num Mod i = 0 Then Return False
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        If EsPrimo(n) Then
            Console.WriteLine(n & " es primo.")
        Else
            Console.WriteLine(n & " no es primo.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 74: Mostrar la tabla de multiplicar de un número

Enunciado:
Pide un número y muestra su tabla de multiplicar del 1 al 10.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To 10
            Console.WriteLine($"{n} x {i} = {n * i}")
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 75: Contar cuántas vocales tiene una frase

Enunciado:
Pide una frase y muestra cuántas vocales contiene.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine().ToLower()
        Dim contador As Integer = 0
        Dim vocales As String = "aeiou"

        For Each c As Char In frase
            If vocales.Contains(c) Then
                contador += 1
            End If
        Next

        Console.WriteLine("Número de vocales: " & contador)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 76: Ordenar tres números de menor a mayor

Enunciado:
Pide tres números y muéstralos ordenados ascendentemente.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim a As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el segundo número: ")
```

```
        Dim b As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el tercer número: ")
        Dim c As Integer = Convert.ToInt32(Console.ReadLine())

        Dim nums As Integer() = {a, b, c}
        Array.Sort(nums)

        Console.WriteLine("Números ordenados: " & String.Join(", ", nums))
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 77: Calcular la suma de los primeros N números naturales

Enunciado:
Pide N y muestra la suma desde 1 hasta N.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Integer = 0

        For i As Integer = 1 To n
            suma += i
        Next

        Console.WriteLine("Suma: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 78: Mostrar los divisores de un número

Enunciado:
Pide un número y muestra todos sus divisores.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine("Divisores:")
        For i As Integer = 1 To n
            If n Mod i = 0 Then
                Console.WriteLine(i)
            End If
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 79: Calcular la media y la desviación estándar de 5 números

Enunciado:
Pide 5 números, calcula y muestra la media y desviación estándar.

Solución:

```
Module Module1
    Sub Main()
        Dim numeros(4) As Double
        For i As Integer = 0 To 4
            Console.Write($"Ingrese número {i + 1}: ")
            numeros(i) = Convert.ToDouble(Console.ReadLine())
        Next

        Dim suma As Double = 0
        For Each num As Double In numeros
            suma += num
        Next
        Dim media As Double = suma / 5

        Dim sumaVarianza As Double = 0
        For Each num As Double In numeros
            sumaVarianza += (num - media) ^ 2
        Next

        Dim desviacion As Double = Math.Sqrt(sumaVarianza / 5)

        Console.WriteLine($"Media: {media}")
        Console.WriteLine($"Desviación estándar: {desviacion}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 80: Convertir grados Celsius a Fahrenheit y Kelvin

Enunciado:
Pide grados Celsius y muestra su equivalente en Fahrenheit y Kelvin.

Solución:

```
Module Module1
    Sub Main()
        Console.Write("Ingrese grados Celsius: ")
        Dim celsius As Double = Convert.ToDouble(Console.ReadLine())

        Dim fahrenheit As Double = celsius * 9 / 5 + 32
        Dim kelvin As Double = celsius + 273.15

        Console.WriteLine($"Fahrenheit: {fahrenheit}")
        Console.WriteLine($"Kelvin: {kelvin}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 81: Mostrar los primeros N números Fibonacci

```
Module Module1
    Sub Main()
        Console.Write("Ingrese N para la serie Fibonacci: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim a As Integer = 0, b As Integer = 1, c As Integer

        For i As Integer = 1 To n
            Console.Write(a & " ")
            c = a + b
            a = b
            b = c
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 82: Contar dígitos de un número

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim contador As Integer = 0
        Dim temp As Integer = Math.Abs(num)

        While temp > 0
            temp \= 10
            contador += 1
        End While

        If num = 0 Then contador = 1
        Console.WriteLine("Número de dígitos: " & contador)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 83: Invertir una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim invertido As String = ""

        For i As Integer = texto.Length - 1 To 0 Step -1
            invertido &= texto(i)
        Next

        Console.WriteLine("Cadena invertida: " & invertido)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 84: Generar N números aleatorios entre 1 y 100

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de números aleatorios: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim rnd As New Random()

        For i As Integer = 1 To n
            Console.WriteLine(rnd.Next(1, 101))
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 85: Calcular el factorial de un número

```
Module Module1
    Function Factorial(n As Integer) As Long
        Dim resultado As Long = 1
        For i As Integer = 2 To n
            resultado *= i
        Next
        Return resultado
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"{num}! = {Factorial(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 86: Comprobar si una palabra es palíndromo

```
Module Module1
    Function EsPalindromo(texto As String) As Boolean
        Dim limpio As String = texto.ToLower().Replace(" ", "")
        Dim invertido As String = StrReverse(limpio)
        Return limpio = invertido
    End Function

    Sub Main()
        Console.Write("Ingrese una palabra o frase: ")
        Dim entrada As String = Console.ReadLine()
        If EsPalindromo(entrada) Then
            Console.WriteLine("Es palíndromo")
        Else
            Console.WriteLine("No es palíndromo")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 87: Calcular la suma de dígitos de un número

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Math.Abs(Convert.ToInt32(Console.ReadLine()))
        Dim suma As Integer = 0

        While num > 0
            suma += num Mod 10
            num \= 10
        End While

        Console.WriteLine("Suma de dígitos: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 88: Contar cuántas veces aparece una letra en una frase

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine().ToLower()
        Console.Write("Ingrese una letra: ")
        Dim letra As Char = Char.ToLower(Console.ReadLine()(0))
        Dim contador As Integer = 0

        For Each c As Char In frase
            If c = letra Then
                contador += 1
            End If
        Next

        Console.WriteLine($"La letra '{letra}' aparece {contador} veces.")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 89: Calcular el promedio de números introducidos hasta que el usuario escriba -1

```
Module Module1
    Sub Main()
        Dim suma As Double = 0
        Dim contador As Integer = 0
        Dim num As Integer

        Do
            Console.Write("Ingrese un número (-1 para terminar): ")
            num = Convert.ToInt32(Console.ReadLine())
            If num <> -1 Then
                suma += num
                contador += 1
            End If
        Loop While num <> -1

        If contador > 0 Then
```

```
            Console.WriteLine("Promedio: " & (suma / contador))
        Else
            Console.WriteLine("No se ingresaron números.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 90: Mostrar un menú con opciones y ejecutar según elección

```
Module Module1
    Sub Main()
        Dim opcion As Integer

        Do
            Console.Clear()
            Console.WriteLine("Menú:")
            Console.WriteLine("1. Saludar")
            Console.WriteLine("2. Mostrar fecha actual")
            Console.WriteLine("3. Salir")
            Console.Write("Seleccione una opción: ")
            opcion = Convert.ToInt32(Console.ReadLine())

            Select Case opcion
                Case 1
                    Console.WriteLine("¡Hola! ¿Cómo estás?")
                Case 2
                    Console.WriteLine("Fecha actual: " &
DateTime.Now.ToString("dd/MM/yyyy"))
                Case 3
                    Console.WriteLine("Saliendo...")
                Case Else
                    Console.WriteLine("Opción inválida.")
            End Select

            If opcion <> 3 Then
                Console.WriteLine("Presione Enter para continuar...")
                Console.ReadLine()
            End If
        Loop While opcion <> 3
    End Sub
End Module
```

Ejercicio 91: Contar cuántos números pares hay en una lista

```
Module Module1
    Sub Main()
        Console.Write("¿Cuántos números quiere ingresar? ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim contadorPares As Integer = 0

        For i As Integer = 1 To n
            Console.Write($"Número {i}: ")
            Dim num As Integer = Convert.ToInt32(Console.ReadLine())
            If num Mod 2 = 0 Then
                contadorPares += 1
            End If
```

```
        Next

        Console.WriteLine($"Cantidad de números pares: {contadorPares}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 92: Mostrar el número mayor y menor de una lista

```
Module Module1
    Sub Main()
        Console.Write("¿Cuántos números quiere ingresar? ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim numeros(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Número {i + 1}: ")
            numeros(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Dim mayor As Integer = numeros.Max()
        Dim menor As Integer = numeros.Min()

        Console.WriteLine($"Mayor: {mayor}")
        Console.WriteLine($"Menor: {menor}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 93: Ingresar palabras hasta que se escriba "fin" y mostrarlas ordenadas

```
Module Module1
    Sub Main()
        Dim lista As New List(Of String)
        Dim palabra As String

        Do
            Console.Write("Ingrese una palabra (\"fin\" para terminar): ")
            palabra = Console.ReadLine()
            If palabra.ToLower() <> "fin" Then
                lista.Add(palabra)
            End If
        Loop While palabra.ToLower() <> "fin"

        lista.Sort()
        Console.WriteLine("Palabras ordenadas:")
        For Each p In lista
            Console.WriteLine(p)
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 94: Mostrar un triángulo de asteriscos con N filas

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el número de filas: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n
            Console.WriteLine(New String("*"c, i))
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 95: Verificar si un número es capicúa (palíndromo numérico)

```
Module Module1
    Function EsCapicua(num As Integer) As Boolean
        Dim str As String = num.ToString()
        Dim invertido As String = StrReverse(str)
        Return str = invertido
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If EsCapicua(num) Then
            Console.WriteLine($"{num} es capicúa.")
        Else
            Console.WriteLine($"{num} no es capicúa.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 96: Mostrar los números primos entre 1 y N

```
Module Module1
    Function EsPrimo(num As Integer) As Boolean
        If num < 2 Then Return False
        For i As Integer = 2 To Math.Sqrt(num)
            If num Mod i = 0 Then Return False
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine("Números primos:")
        For i As Integer = 2 To n
            If EsPrimo(i) Then
                Console.Write(i & " ")
            End If
```

```
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 97: Calcular el área de un círculo dado su radio

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el radio del círculo: ")
        Dim radio As Double = Convert.ToDouble(Console.ReadLine())
        Dim area As Double = Math.PI * radio * radio

        Console.WriteLine("Área: " & area)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 98: Convertir un número decimal a binario

```
Module Module1
    Function DecimalABinario(num As Integer) As String
        If num = 0 Then Return "0"
        Dim binario As String = ""
        Dim n As Integer = num

        While n > 0
            binario = (n Mod 2).ToString() & binario
            n \= 2
        End While

        Return binario
    End Function

    Sub Main()
        Console.Write("Ingrese un número decimal: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"Binario: {DecimalABinario(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 99: Mostrar los primeros N números pares

```
Module Module1
    Sub Main()
        Console.Write("Ingrese N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim contador As Integer = 0
        Dim num As Integer = 2

        While contador < n
            Console.Write(num & " ")
            num += 2
```

```
            contador += 1
        End While

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 100: Mostrar una pirámide numérica con N filas

```
Module Module1
    Sub Main()
        Console.Write("Ingrese número de filas: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n
            For j As Integer = 1 To i
                Console.Write(j & " ")
            Next
            Console.WriteLine()
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 101: Calcular la suma de números pares entre 1 y N

```
Module Module1
    Sub Main()
        Console.Write("Ingrese N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Integer = 0

        For i As Integer = 2 To n Step 2
            suma += i
        Next

        Console.WriteLine("Suma de pares: " & suma)
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 102: Determinar si un año es bisiesto

```
Module Module1
    Function EsBisiesto(año As Integer) As Boolean
        Return (año Mod 4 = 0 AndAlso año Mod 100 <> 0) OrElse (año Mod 400 = 0)
    End Function

    Sub Main()
        Console.Write("Ingrese un año: ")
        Dim año As Integer = Convert.ToInt32(Console.ReadLine())

        If EsBisiesto(año) Then
            Console.WriteLine($"{año} es bisiesto.")
        Else
```

```
            Console.WriteLine($"{año} no es bisiesto.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 103: Calcular la media y la desviación estándar de N números

```
Module Module1
    Sub Main()
        Console.Write("¿Cuántos números va a ingresar? ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim nums(n - 1) As Double
        Dim suma As Double = 0

        For i As Integer = 0 To n - 1
            Console.Write($"Número {i + 1}: ")
            nums(i) = Convert.ToDouble(Console.ReadLine())
            suma += nums(i)
        Next

        Dim media As Double = suma / n
        Dim sumaVar As Double = 0

        For Each num In nums
            sumaVar += (num - media) ^ 2
        Next

        Dim desviacion As Double = Math.Sqrt(sumaVar / n)

        Console.WriteLine($"Media: {media}")
        Console.WriteLine($"Desviación estándar: {desviacion}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 104: Generar y mostrar una matriz N x M con números aleatorios entre 1 y 50

```
Module Module1
    Sub Main()
        Console.Write("Ingrese número de filas (N): ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese número de columnas (M): ")
        Dim m As Integer = Convert.ToInt32(Console.ReadLine())
        Dim rnd As New Random()
        Dim matriz(n - 1, m - 1) As Integer

        For i As Integer = 0 To n - 1
            For j As Integer = 0 To m - 1
                matriz(i, j) = rnd.Next(1, 51)
            Next
        Next

        Console.WriteLine("Matriz generada:")
        For i As Integer = 0 To n - 1
            For j As Integer = 0 To m - 1
```

```
                Console.Write(matriz(i, j).ToString().PadLeft(3) & " ")
            Next
            Console.WriteLine()
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 105: Transponer una matriz cuadrada N x N

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño N de la matriz cuadrada: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim matriz(n - 1, n - 1) As Integer
        Dim transpuesta(n - 1, n - 1) As Integer
        Dim rnd As New Random()

        For i As Integer = 0 To n - 1
            For j As Integer = 0 To n - 1
                matriz(i, j) = rnd.Next(1, 100)
            Next
        Next

        Console.WriteLine("Matriz original:")
        For i As Integer = 0 To n - 1
            For j As Integer = 0 To n - 1
                Console.Write(matriz(i, j).ToString().PadLeft(4))
            Next
            Console.WriteLine()
        Next

        For i As Integer = 0 To n - 1
            For j As Integer = 0 To n - 1
                transpuesta(j, i) = matriz(i, j)
            Next
        Next

        Console.WriteLine("Matriz transpuesta:")
        For i As Integer = 0 To n - 1
            For j As Integer = 0 To n - 1
                Console.Write(transpuesta(i, j).ToString().PadLeft(4))
            Next
            Console.WriteLine()
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 106: Contar vocales y consonantes en una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine().ToLower()
        Dim vocales As Integer = 0
```

```
        Dim consonantes As Integer = 0

        For Each c As Char In texto
            If Char.IsLetter(c) Then
                If "aeiou".Contains(c) Then
                    vocales += 1
                Else
                    consonantes += 1
                End If
            End If
        Next

        Console.WriteLine($"Vocales: {vocales}")
        Console.WriteLine($"Consonantes: {consonantes}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 107: Convertir grados Celsius a Fahrenheit y Kelvin

```
Module Module1
    Sub Main()
        Console.Write("Ingrese temperatura en °C: ")
        Dim celsius As Double = Convert.ToDouble(Console.ReadLine())
        Dim fahrenheit As Double = celsius * 9 / 5 + 32
        Dim kelvin As Double = celsius + 273.15

        Console.WriteLine($"Fahrenheit: {fahrenheit}")
        Console.WriteLine($"Kelvin: {kelvin}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 108: Generar una lista de N números y ordenarla descendentemente

```
Module Module1
    Sub Main()
        Console.Write("Ingrese cantidad N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim lista As New List(Of Integer)

        For i As Integer = 1 To n
            Console.Write($"Número {i}: ")
            lista.Add(Convert.ToInt32(Console.ReadLine()))
        Next

        lista.Sort()
        lista.Reverse()

        Console.WriteLine("Lista ordenada descendentemente:")
        For Each num In lista
            Console.WriteLine(num)
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 109: Mostrar la tabla de multiplicar de un número

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To 10
            Console.WriteLine($"{num} x {i} = {num * i}")
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 110: Detectar si una cadena contiene sólo números

```
Module Module1
    Function EsNumero(cadena As String) As Boolean
        For Each c As Char In cadena
            If Not Char.IsDigit(c) Then
                Return False
            End If
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()

        If EsNumero(texto) Then
            Console.WriteLine("La cadena contiene sólo números.")
        Else
            Console.WriteLine("La cadena NO contiene sólo números.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 111: Calcular el factorial de un número usando recursión

```
Module Module1
    Function Factorial(n As Integer) As Long
        If n <= 1 Then
            Return 1
        Else
            Return n * Factorial(n - 1)
        End If
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
```

```
        Console.WriteLine($"Factorial de {num} es {Factorial(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 112: Invertir una cadena de texto

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim invertido As String = StrReverse(texto)

        Console.WriteLine($"Cadena invertida: {invertido}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 113: Calcular la suma de dígitos de un número

```
Module Module1
    Function SumaDigitos(num As Integer) As Integer
        Dim suma As Integer = 0
        While num > 0
            suma += num Mod 10
            num \= 10
        End While
        Return suma
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"Suma de dígitos: {SumaDigitos(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 114: Validar si un email es correcto (básico)

```
Module Module1
    Function EsEmailValido(email As String) As Boolean
        Return email.Contains("@") AndAlso email.Contains(".")
    End Function

    Sub Main()
        Console.Write("Ingrese un email: ")
        Dim email As String = Console.ReadLine()

        If EsEmailValido(email) Then
            Console.WriteLine("Email válido (básico).")
        Else
            Console.WriteLine("Email inválido.")
        End If
```

```
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 115: Mostrar números Fibonacci hasta un límite N

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el límite N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim a As Integer = 0, b As Integer = 1, c As Integer = 0

        Console.Write("Serie Fibonacci: ")
        While a <= n
            Console.Write(a & " ")
            c = a + b
            a = b
            b = c
        End While

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 116: Calcular la media de números pares ingresados hasta que se ingrese un número impar

```
Module Module1
    Sub Main()
        Dim suma As Integer = 0
        Dim contador As Integer = 0
        Dim num As Integer

        Do
            Console.Write("Ingrese un número (impar para terminar): ")
            num = Convert.ToInt32(Console.ReadLine())
            If num Mod 2 = 0 Then
                suma += num
                contador += 1
            End If
        Loop While num Mod 2 = 0

        If contador > 0 Then
            Console.WriteLine($"Media de números pares: {suma / contador}")
        Else
            Console.WriteLine("No se ingresaron números pares.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 117: Contar cuántas palabras tiene una frase

```
Module Module1
```

```
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine().Trim()
        Dim palabras As String() = frase.Split(New Char() {" "c},
StringSplitOptions.RemoveEmptyEntries)

        Console.WriteLine($"Número de palabras: {palabras.Length}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 118: Calcular el mínimo común múltiplo (MCM) de dos números

```
Module Module1
    Function MCD(a As Integer, b As Integer) As Integer
        While b <> 0
            Dim temp As Integer = b
            b = a Mod b
            a = temp
        End While
        Return a
    End Function

    Function MCM(a As Integer, b As Integer) As Integer
        Return (a * b) \ MCD(a, b)
    End Function

    Sub Main()
        Console.Write("Ingrese primer número: ")
        Dim a As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese segundo número: ")
        Dim b As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"MCM de {a} y {b} es {MCM(a, b)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 119: Generar N números aleatorios únicos entre 1 y 100

```
Module Module1
    Sub Main()
        Console.Write("Ingrese N: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim rnd As New Random()
        Dim numeros As New HashSet(Of Integer)

        While numeros.Count < n
            numeros.Add(rnd.Next(1, 101))
        End While

        Console.WriteLine("Números generados:")
        For Each num In numeros
            Console.WriteLine(num)
        Next

        Console.ReadLine()
    End Sub
```

```
End Module
```

Ejercicio 120: Mostrar los divisores de un número

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"Divisores de {num}:")
        For i As Integer = 1 To num
            If num Mod i = 0 Then
                Console.Write(i & " ")
            End If
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 121: Convertir número decimal a binario

```
Module Module1
    Function DecimalABinario(n As Integer) As String
        Dim binario As String = ""
        While n > 0
            binario = (n Mod 2).ToString() & binario
            n \= 2
        End While
        Return If(binario = "", "0", binario)
    End Function

    Sub Main()
        Console.Write("Ingrese un número decimal: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"Binario: {DecimalABinario(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 122: Calcular la potencia de un número (base y exponente) sin usar Math.Pow

```
Module Module1
    Function Potencia(base As Double, exponente As Integer) As Double
        Dim resultado As Double = 1
        For i As Integer = 1 To Math.Abs(exponente)
            resultado *= base
        Next
        If exponente < 0 Then
            resultado = 1 / resultado
        End If
        Return resultado
    End Function
```

```
    Sub Main()
        Console.Write("Ingrese base: ")
        Dim base As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese exponente (entero): ")
        Dim exponente As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"{base} ^ {exponente} = {Potencia(base, exponente)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 123: Encontrar el elemento máximo y mínimo en una lista

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de números: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim lista As New List(Of Integer)

        For i As Integer = 1 To n
            Console.Write($"Número {i}: ")
            lista.Add(Convert.ToInt32(Console.ReadLine()))
        Next

        Dim maximo As Integer = lista.Max()
        Dim minimo As Integer = lista.Min()

        Console.WriteLine($"Máximo: {maximo}")
        Console.WriteLine($"Mínimo: {minimo}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 124: Calcular la suma de una serie aritmética (N términos)

```
Module Module1
    Function SumaAritmetica(a1 As Double, d As Double, n As Integer) As Double
        Return (n / 2) * (2 * a1 + (n - 1) * d)
    End Function

    Sub Main()
        Console.Write("Ingrese primer término (a1): ")
        Dim a1 As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese diferencia común (d): ")
        Dim d As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese número de términos (n): ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"Suma de la serie aritmética: {SumaAritmetica(a1, d,
n)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 125: Contar letras mayúsculas y minúsculas en una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim mayusculas As Integer = 0
        Dim minusculas As Integer = 0

        For Each c As Char In texto
            If Char.IsUpper(c) Then
                mayusculas += 1
            ElseIf Char.IsLower(c) Then
                minusculas += 1
            End If
        Next

        Console.WriteLine($"Mayúsculas: {mayusculas}")
        Console.WriteLine($"Minúsculas: {minusculas}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 126: Comprobar si un número es primo

```
Module Module1
    Function EsPrimo(n As Integer) As Boolean
        If n <= 1 Then Return False
        For i As Integer = 2 To Math.Sqrt(n)
            If n Mod i = 0 Then Return False
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If EsPrimo(num) Then
            Console.WriteLine($"{num} es primo.")
        Else
            Console.WriteLine($"{num} no es primo.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 127: Mostrar una pirámide de asteriscos

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la altura de la pirámide: ")
        Dim altura As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To altura
            Console.WriteLine(New String(" "c, altura - i) & New String("*"c, 2
* i - 1))
```

```
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 128: Ordenar un arreglo usando el método de burbuja

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        ' Método burbuja
        For i As Integer = 0 To n - 2
            For j As Integer = 0 To n - 2 - i
                If arr(j) > arr(j + 1) Then
                    Dim temp As Integer = arr(j)
                    arr(j) = arr(j + 1)
                    arr(j + 1) = temp
                End If
            Next
        Next

        Console.WriteLine("Arreglo ordenado:")
        For Each num In arr
            Console.WriteLine(num)
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 129: Convertir número binario a decimal

```
Module Module1
    Function BinarioADecimal(binario As String) As Integer
        Dim decimalNum As Integer = 0
        For i As Integer = 0 To binario.Length - 1
            If binario(binario.Length - 1 - i) = "1"c Then
                decimalNum += 2 ^ i
            End If
        Next
        Return decimalNum
    End Function

    Sub Main()
        Console.Write("Ingrese número binario: ")
        Dim binario As String = Console.ReadLine()

        Console.WriteLine($"Decimal: {BinarioADecimal(binario)}")
        Console.ReadLine()
```

```
    End Sub
End Module
```

Ejercicio 130: Calcular el área y perímetro de un rectángulo

```
Module Module1
    Sub Main()
        Console.Write("Ingrese base: ")
        Dim base As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese altura: ")
        Dim altura As Double = Convert.ToDouble(Console.ReadLine())

        Dim area As Double = base * altura
        Dim perimetro As Double = 2 * (base + altura)

        Console.WriteLine($"Área: {area}")
        Console.WriteLine($"Perímetro: {perimetro}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 131: Calcular el número de dígitos de un número entero

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim contador As Integer = 0
        Dim temp As Integer = Math.Abs(num)

        If temp = 0 Then contador = 1

        While temp > 0
            temp \= 10
            contador += 1
        End While

        Console.WriteLine($"Número de dígitos: {contador}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 132: Reemplazar todas las vocales en una cadena por un carácter dado

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Console.Write("Ingrese el carácter de reemplazo: ")
        Dim reemplazo As Char = Console.ReadLine()(0)

        Dim vocales As String = "aeiouAEIOU"
        Dim resultado As String = ""
```

```
        For Each c As Char In texto
            If vocales.Contains(c) Then
                resultado &= reemplazo
            Else
                resultado &= c
            End If
        Next

        Console.WriteLine($"Cadena resultante: {resultado}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 133: Mostrar los primeros N números primos

```
Module Module1
    Function EsPrimo(n As Integer) As Boolean
        If n <= 1 Then Return False
        For i As Integer = 2 To Math.Sqrt(n)
            If n Mod i = 0 Then Return False
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese la cantidad de números primos a mostrar: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim contador As Integer = 0
        Dim num As Integer = 2

        While contador < n
            If EsPrimo(num) Then
                Console.Write(num & " ")
                contador += 1
            End If
            num += 1
        End While

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 134: Contar caracteres alfabéticos, dígitos y otros en una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()

        Dim letras As Integer = 0
        Dim digitos As Integer = 0
        Dim otros As Integer = 0

        For Each c As Char In texto
            If Char.IsLetter(c) Then
                letras += 1
            ElseIf Char.IsDigit(c) Then
                digitos += 1
```

```
            Else
                otros += 1
            End If
        Next

        Console.WriteLine($"Letras: {letras}")
        Console.WriteLine($"Dígitos: {digitos}")
        Console.WriteLine($"Otros caracteres: {otros}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 135: Generar una matriz identidad de tamaño N x N

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el tamaño de la matriz: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim matriz(n - 1, n - 1) As Integer

        For i As Integer = 0 To n - 1
            For j As Integer = 0 To n - 1
                matriz(i, j) = If(i = j, 1, 0)
            Next
        Next

        Console.WriteLine("Matriz identidad:")
        For i As Integer = 0 To n - 1
            For j As Integer = 0 To n - 1
                Console.Write(matriz(i, j) & " ")
            Next
            Console.WriteLine()
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 136: Calcular la suma de números pares en una lista

```
Module Module1
    Sub Main()
        Console.Write("Ingrese cantidad de números: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim lista As New List(Of Integer)

        For i As Integer = 1 To n
            Console.Write($"Número {i}: ")
            lista.Add(Convert.ToInt32(Console.ReadLine()))
        Next

        Dim sumaPares As Integer = lista.Where(Function(x) x Mod 2 = 0).Sum()

        Console.WriteLine($"Suma de números pares: {sumaPares}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 137: Determinar si una cadena es un palíndromo (ignorar mayúsculas y espacios)

```
Module Module1
    Function EsPalindromo(texto As String) As Boolean
        Dim limpio As String = New String(texto.Where(Function(c)
Char.IsLetterOrDigit(c)).ToArray()).ToLower()
        Dim invertido As String = StrReverse(limpio)
        Return limpio = invertido
    End Function

    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()

        If EsPalindromo(texto) Then
            Console.WriteLine("Es un palíndromo.")
        Else
            Console.WriteLine("No es un palíndromo.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 138: Encontrar el segundo número más grande en un arreglo

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Dim max1 As Integer = Integer.MinValue
        Dim max2 As Integer = Integer.MinValue

        For Each num In arr
            If num > max1 Then
                max2 = max1
                max1 = num
            ElseIf num > max2 And num <> max1 Then
                max2 = num
            End If
        Next

        If max2 = Integer.MinValue Then
            Console.WriteLine("No hay segundo número más grande distinto.")
        Else
            Console.WriteLine($"Segundo número más grande: {max2}")
        End If

        Console.ReadLine()
    End Sub
```

```
End Module
```

Ejercicio 139: Mostrar tabla de multiplicar de un número hasta 12

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To 12
            Console.WriteLine($"{num} x {i} = {num * i}")
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 140: Eliminar espacios extras en una cadena (dejar solo uno entre palabras)

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim palabras As String() = texto.Split(New Char() {" "c},
StringSplitOptions.RemoveEmptyEntries)
        Dim resultado As String = String.Join(" ", palabras)

        Console.WriteLine($"Cadena sin espacios extras: {resultado}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 141: Calcular el factorial de un número usando recursión

```
Module Module1
    Function Factorial(n As Integer) As Long
        If n <= 1 Then
            Return 1
        Else
            Return n * Factorial(n - 1)
        End If
    End Function

    Sub Main()
        Console.Write("Ingrese un número para calcular factorial: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"Factorial de {num} = {Factorial(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 142: Invertir una cadena sin usar funciones incorporadas

```
Module Module1
    Function InvertirCadena(cadena As String) As String
        Dim resultado As String = ""
        For i As Integer = cadena.Length - 1 To 0 Step -1
            resultado &= cadena(i)
        Next
        Return resultado
    End Function

    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()

        Console.WriteLine($"Cadena invertida: {InvertirCadena(texto)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 143: Calcular el máximo común divisor (MCD) usando algoritmo de Euclides

```
Module Module1
    Function MCD(a As Integer, b As Integer) As Integer
        While b <> 0
            Dim temp As Integer = b
            b = a Mod b
            a = temp
        End While
        Return a
    End Function

    Sub Main()
        Console.Write("Ingrese primer número: ")
        Dim num1 As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese segundo número: ")
        Dim num2 As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"MCD({num1}, {num2}) = {MCD(num1, num2)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 144: Mostrar números Fibonacci hasta un límite dado

```
Module Module1
    Sub Main()
        Console.Write("Ingrese límite para Fibonacci: ")
        Dim limite As Integer = Convert.ToInt32(Console.ReadLine())
        Dim a As Integer = 0
        Dim b As Integer = 1

        Console.Write($"{a} {b} ")
        Dim c As Integer = a + b

        While c <= limite
            Console.Write($"{c} ")
```

```
            a = b
            b = c
            c = a + b
        End While

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 145: Determinar si un año es bisiesto

```
Module Module1
    Function EsBisiesto(anio As Integer) As Boolean
        Return (anio Mod 400 = 0) Or ((anio Mod 4 = 0) And (anio Mod 100 <> 0))
    End Function

    Sub Main()
        Console.Write("Ingrese un año: ")
        Dim anio As Integer = Convert.ToInt32(Console.ReadLine())

        If EsBisiesto(anio) Then
            Console.WriteLine($"{anio} es un año bisiesto.")
        Else
            Console.WriteLine($"{anio} no es un año bisiesto.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 146: Contar cuántas palabras hay en una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim palabras As String() = texto.Split(New Char() {" "c},
StringSplitOptions.RemoveEmptyEntries)

        Console.WriteLine($"Número de palabras: {palabras.Length}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 147: Calcular la media aritmética de números ingresados por el usuario

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de números: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Double = 0

        For i As Integer = 1 To n
            Console.Write($"Número {i}: ")
```

```
            suma += Convert.ToDouble(Console.ReadLine())
        Next

        Console.WriteLine($"Media aritmética: {suma / n}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 148: Buscar un elemento en un arreglo y mostrar su posición (si existe)

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Console.Write("Ingrese número a buscar: ")
        Dim buscado As Integer = Convert.ToInt32(Console.ReadLine())
        Dim pos As Integer = Array.IndexOf(arr, buscado)

        If pos >= 0 Then
            Console.WriteLine($"Número encontrado en posición {pos}")
        Else
            Console.WriteLine("Número no encontrado.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 149: Calcular el promedio de notas y determinar si aprueba (promedio >= 6)

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de notas: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Double = 0

        For i As Integer = 1 To n
            Console.Write($"Nota {i}: ")
            suma += Convert.ToDouble(Console.ReadLine())
        Next

        Dim promedio As Double = suma / n
        Console.WriteLine($"Promedio: {promedio:F2}")

        If promedio >= 6 Then
            Console.WriteLine("Aprobado")
        Else
            Console.WriteLine("Reprobado")
        End If
```

```
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 150: Imprimir los números pares entre dos números dados

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el primer número: ")
        Dim inicio As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese el segundo número: ")
        Dim fin As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = inicio To fin
            If i Mod 2 = 0 Then
                Console.Write(i & " ")
            End If
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 151: Contar cuántas veces aparece un carácter en una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Console.Write("Ingrese el carácter a contar: ")
        Dim c As Char = Console.ReadLine()(0)

        Dim contador As Integer = 0
        For Each ch As Char In texto
            If ch = c Then contador += 1
        Next

        Console.WriteLine($"El carácter '{c}' aparece {contador} veces.")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 152: Calcular el número de palabras que comienzan con una letra específica

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Console.Write("Ingrese la letra inicial: ")
        Dim letra As Char = Char.ToLower(Console.ReadLine()(0))

        Dim palabras As String() = texto.Split(New Char() {" "c},
StringSplitOptions.RemoveEmptyEntries)
        Dim contador As Integer = 0
```

```
        For Each palabra As String In palabras
            If Char.ToLower(palabra(0)) = letra Then
                contador += 1
            End If
        Next

        Console.WriteLine($"Número de palabras que comienzan con '{letra}':
{contador}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 153: Encontrar el número más pequeño y el más grande en un arreglo

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Dim min As Integer = arr.Min()
        Dim max As Integer = arr.Max()

        Console.WriteLine($"Número más pequeño: {min}")
        Console.WriteLine($"Número más grande: {max}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 154: Verificar si un número es perfecto (suma de divisores = número)

```
Module Module1
    Function EsPerfecto(num As Integer) As Boolean
        Dim suma As Integer = 0
        For i As Integer = 1 To num \ 2
            If num Mod i = 0 Then suma += i
        Next
        Return suma = num
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If EsPerfecto(num) Then
            Console.WriteLine($"{num} es un número perfecto.")
        Else
            Console.WriteLine($"{num} no es un número perfecto.")
        End If

        Console.ReadLine()
    End Sub
```

```
End Module
```

Ejercicio 155: Calcular el área de un triángulo dados sus tres lados (fórmula de Herón)

```
Module Module1
    Sub Main()
        Console.Write("Ingrese lado 1: ")
        Dim a As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese lado 2: ")
        Dim b As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese lado 3: ")
        Dim c As Double = Convert.ToDouble(Console.ReadLine())

        Dim s As Double = (a + b + c) / 2
        Dim area As Double = Math.Sqrt(s * (s - a) * (s - b) * (s - c))

        Console.WriteLine($"Área del triángulo: {area:F2}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 156: Mostrar números múltiplos de 3 entre 1 y N

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el límite superior: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n
            If i Mod 3 = 0 Then
                Console.Write(i & " ")
            End If
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 157: Convertir una temperatura de Celsius a Fahrenheit

```
Module Module1
    Sub Main()
        Console.Write("Ingrese temperatura en Celsius: ")
        Dim celsius As Double = Convert.ToDouble(Console.ReadLine())

        Dim fahrenheit As Double = (celsius * 9 / 5) + 32
        Console.WriteLine($"Temperatura en Fahrenheit: {fahrenheit:F2}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 158: Generar una pirámide de asteriscos con N niveles

```
Module Module1
    Sub Main()
        Console.Write("Ingrese número de niveles: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n
            Console.WriteLine(New String(" "c, n - i) & New String("*"c, 2 * i -
1))
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 159: Calcular la suma de dígitos de un número

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Integer = 0
        Dim temp As Integer = Math.Abs(num)

        While temp > 0
            suma += temp Mod 10
            temp \= 10
        End While

        Console.WriteLine($"Suma de dígitos: {suma}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 160: Ordenar un arreglo de números enteros de menor a mayor

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Array.Sort(arr)

        Console.WriteLine("Arreglo ordenado:")
        For Each num In arr
            Console.Write(num & " ")
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 161: Contar cuántos números negativos hay en un arreglo

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Dim negativos As Integer = 0
        For Each num In arr
            If num < 0 Then negativos += 1
        Next

        Console.WriteLine($"Cantidad de números negativos: {negativos}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 162: Validar si una cadena es un número entero positivo

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim esEnteroPositivo As Boolean = Integer.TryParse(texto, Nothing)
AndAlso Convert.ToInt32(texto) > 0

        If esEnteroPositivo Then
            Console.WriteLine("La cadena es un número entero positivo.")
        Else
            Console.WriteLine("La cadena NO es un número entero positivo.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 163: Calcular el número de días entre dos fechas

```
Module Module1
    Sub Main()
        Console.Write("Ingrese fecha inicial (dd/mm/aaaa): ")
        Dim fecha1 As Date = Date.Parse(Console.ReadLine())
        Console.Write("Ingrese fecha final (dd/mm/aaaa): ")
        Dim fecha2 As Date = Date.Parse(Console.ReadLine())

        Dim dias As Integer = Math.Abs((fecha2 - fecha1).Days)
        Console.WriteLine($"Días entre fechas: {dias}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 164: Convertir un número decimal a binario

```
Module Module1
    Function DecimalABinario(num As Integer) As String
        If num = 0 Then Return "0"
        Dim binario As String = ""

        Dim n As Integer = num
        While n > 0
            binario = (n Mod 2).ToString() & binario
            n \= 2
        End While

        Return binario
    End Function

    Sub Main()
        Console.Write("Ingrese un número decimal: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"Binario: {DecimalABinario(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 165: Mostrar los números primos entre dos números dados

```
Module Module1
    Function EsPrimo(num As Integer) As Boolean
        If num < 2 Then Return False
        For i As Integer = 2 To Math.Sqrt(num)
            If num Mod i = 0 Then Return False
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese número inicial: ")
        Dim inicio As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese número final: ")
        Dim fin As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine("Números primos:")
        For i As Integer = inicio To fin
            If EsPrimo(i) Then Console.Write(i & " ")
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 166: Calcular el promedio ponderado de notas con pesos distintos

```
Module Module1
    Sub Main()
        Console.Write("Ingrese cantidad de notas: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
```

```
        Dim sumaNotas As Double = 0
        Dim sumaPesos As Double = 0

        For i As Integer = 1 To n
            Console.Write($"Nota {i}: ")
            Dim nota As Double = Convert.ToDouble(Console.ReadLine())
            Console.Write($"Peso {i}: ")
            Dim peso As Double = Convert.ToDouble(Console.ReadLine())

            sumaNotas += nota * peso
            sumaPesos += peso
        Next

        Dim promedioPonderado As Double = sumaNotas / sumaPesos
        Console.WriteLine($"Promedio ponderado: {promedioPonderado:F2}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 167: Reemplazar todas las vocales en una cadena por '*'

```
Module Module1
    Function ReemplazarVocales(texto As String) As String
        Dim vocales As String = "aeiouAEIOU"
        Dim resultado As String = ""

        For Each c As Char In texto
            If vocales.Contains(c) Then
                resultado &= "*"
            Else
                resultado &= c
            End If
        Next

        Return resultado
    End Function

    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Console.WriteLine(ReemplazarVocales(texto))
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 168: Calcular la suma de los primeros N números pares

```
Module Module1
    Sub Main()
        Console.Write("Ingrese cantidad de números pares: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Integer = 0

        For i As Integer = 1 To n
            suma += 2 * i
        Next

        Console.WriteLine($"Suma de los primeros {n} números pares: {suma}")
```

```
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 169: Mostrar un menú simple con opciones para sumar o restar dos números

```
Module Module1
    Sub Main()
        Console.WriteLine("Menú:")
        Console.WriteLine("1. Sumar dos números")
        Console.WriteLine("2. Restar dos números")
        Console.Write("Seleccione una opción: ")
        Dim opcion As Integer = Convert.ToInt32(Console.ReadLine())

        Console.Write("Ingrese primer número: ")
        Dim num1 As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese segundo número: ")
        Dim num2 As Double = Convert.ToDouble(Console.ReadLine())

        Select Case opcion
            Case 1
                Console.WriteLine($"Suma: {num1 + num2}")
            Case 2
                Console.WriteLine($"Resta: {num1 - num2}")
            Case Else
                Console.WriteLine("Opción inválida.")
        End Select

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 170: Calcular el promedio de un arreglo de números decimales

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Double

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToDouble(Console.ReadLine())
        Next

        Dim suma As Double = arr.Sum()
        Dim promedio As Double = suma / n

        Console.WriteLine($"Promedio: {promedio:F2}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 171: Verificar si una palabra es palíndromo (ignorar mayúsculas y espacios)

```
Module Module1
    Function EsPalindromo(texto As String) As Boolean
        Dim limpio As String = New String(texto.ToLower().Where(Function(c)
Char.IsLetterOrDigit(c)).ToArray())
        Dim reverso As String = New String(limpio.Reverse().ToArray())
        Return limpio = reverso
    End Function

    Sub Main()
        Console.Write("Ingrese una palabra o frase: ")
        Dim texto As String = Console.ReadLine()
        If EsPalindromo(texto) Then
            Console.WriteLine("Es un palíndromo.")
        Else
            Console.WriteLine("No es un palíndromo.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 172: Encontrar el segundo número más grande en un arreglo

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Dim max1 As Integer = Integer.MinValue
        Dim max2 As Integer = Integer.MinValue

        For Each num In arr
            If num > max1 Then
                max2 = max1
                max1 = num
            ElseIf num > max2 AndAlso num < max1 Then
                max2 = num
            End If
        Next

        If max2 = Integer.MinValue Then
            Console.WriteLine("No hay segundo número más grande válido.")
        Else
            Console.WriteLine($"El segundo número más grande es: {max2}")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 173: Contar la cantidad de dígitos en un número entero

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número entero: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim contador As Integer = 0
        Dim temp As Integer = Math.Abs(num)

        If temp = 0 Then contador = 1
        While temp > 0
            temp \= 10
            contador += 1
        End While

        Console.WriteLine($"Cantidad de dígitos: {contador}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 174: Calcular la serie de Fibonacci hasta N términos

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de términos: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim a As Integer = 0
        Dim b As Integer = 1
        Console.Write("Serie Fibonacci: ")

        For i As Integer = 1 To n
            Console.Write(a & " ")
            Dim temp As Integer = a
            a = b
            b = temp + b
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 175: Invertir una cadena sin usar funciones auxiliares

```
Module Module1
    Function InvertirCadena(texto As String) As String
        Dim resultado As String = ""
        For i As Integer = texto.Length - 1 To 0 Step -1
            resultado &= texto(i)
        Next
        Return resultado
    End Function

    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Console.WriteLine("Cadena invertida: " & InvertirCadena(texto))
        Console.ReadLine()
    End Sub
```

```
End Module
```

Ejercicio 176: Mostrar la tabla de multiplicar de un número

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"Tabla de multiplicar del {num}:")
        For i As Integer = 1 To 10
            Console.WriteLine($"{num} x {i} = {num * i}")
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 177: Calcular la media y la mediana de un arreglo de números

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Double

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToDouble(Console.ReadLine())
        Next

        Dim media As Double = arr.Average()
        Array.Sort(arr)
        Dim mediana As Double
        If n Mod 2 = 0 Then
            mediana = (arr(n \ 2 - 1) + arr(n \ 2)) / 2
        Else
            mediana = arr(n \ 2)
        End If

        Console.WriteLine($"Media: {media:F2}")
        Console.WriteLine($"Mediana: {mediana:F2}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 178: Eliminar duplicados de un arreglo de enteros

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
```

```
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Dim sinDuplicados = arr.Distinct().ToArray()

        Console.WriteLine("Arreglo sin duplicados:")
        For Each num In sinDuplicados
            Console.Write(num & " ")
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 179: Contar cuántas palabras tiene una frase

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim texto As String = Console.ReadLine()
        Dim palabras As String() = texto.Split(New Char() {" "c},
StringSplitOptions.RemoveEmptyEntries)
        Console.WriteLine($"Cantidad de palabras: {palabras.Length}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 180: Calcular el factorial de un número usando recursión

```
Module Module1
    Function Factorial(n As Integer) As Long
        If n <= 1 Then
            Return 1
        Else
            Return n * Factorial(n - 1)
        End If
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"Factorial de {num} es: {Factorial(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 181: Determinar si un número es un número perfecto

```
Module Module1
    Function EsPerfecto(num As Integer) As Boolean
        Dim suma As Integer = 0
        For i As Integer = 1 To num \ 2
            If num Mod i = 0 Then suma += i
        Next
        Return suma = num
    End Function
```

```
    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        If EsPerfecto(num) Then
            Console.WriteLine($"{num} es un número perfecto.")
        Else
            Console.WriteLine($"{num} NO es un número perfecto.")
        End If
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 182: Convertir un número decimal a hexadecimal

```
Module Module1
    Function DecimalAHexadecimal(num As Integer) As String
        Return num.ToString("X")
    End Function

    Sub Main()
        Console.Write("Ingrese un número decimal: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"Hexadecimal: {DecimalAHexadecimal(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 183: Contar la cantidad de palabras que empiezan con una letra dada

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una frase: ")
        Dim frase As String = Console.ReadLine()
        Console.Write("Ingrese la letra inicial: ")
        Dim letra As Char = Char.ToLower(Console.ReadLine()(0))

        Dim palabras = frase.Split(New Char() {" "c},
StringSplitOptions.RemoveEmptyEntries)
        Dim contador As Integer = 0

        For Each palabra In palabras
            If Char.ToLower(palabra(0)) = letra Then contador += 1
        Next

        Console.WriteLine($"Cantidad de palabras que empiezan con '{letra}':
{contador}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 184: Calcular la suma de los dígitos de un número entero

```
Module Module1
    Function SumaDigitos(num As Integer) As Integer
        Dim suma As Integer = 0
        Dim n As Integer = Math.Abs(num)
```

```
        While n > 0
            suma += n Mod 10
            n \= 10
        End While

        Return suma
    End Function

    Sub Main()
        Console.Write("Ingrese un número entero: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"Suma de sus dígitos: {SumaDigitos(num)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 185: Mostrar un triángulo de asteriscos de altura N

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la altura del triángulo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n
            Console.WriteLine(New String("*"c, i))
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 186: Validar si una contraseña cumple con ciertos criterios (mayúsculas, minúsculas, dígitos)

```
Module Module1
    Function EsValida(pass As String) As Boolean
        Dim tieneMayus As Boolean = pass.Any(Function(c) Char.IsUpper(c))
        Dim tieneMinus As Boolean = pass.Any(Function(c) Char.IsLower(c))
        Dim tieneDigito As Boolean = pass.Any(Function(c) Char.IsDigit(c))
        Return tieneMayus AndAlso tieneMinus AndAlso tieneDigito AndAlso
pass.Length >= 8
    End Function

    Sub Main()
        Console.Write("Ingrese una contraseña: ")
        Dim password As String = Console.ReadLine()

        If EsValida(password) Then
            Console.WriteLine("La contraseña es válida.")
        Else
            Console.WriteLine("La contraseña NO cumple los requisitos.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 187: Calcular el máximo común divisor (MCD) usando el algoritmo de Euclides

```
Module Module1
    Function MCD(a As Integer, b As Integer) As Integer
        While b <> 0
            Dim temp As Integer = b
            b = a Mod b
            a = temp
        End While
        Return a
    End Function

    Sub Main()
        Console.Write("Ingrese primer número: ")
        Dim num1 As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Ingrese segundo número: ")
        Dim num2 As Integer = Convert.ToInt32(Console.ReadLine())

        Console.WriteLine($"MCD: {MCD(num1, num2)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 188: Mostrar los primeros N números pares

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de números pares a mostrar: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())

        For i As Integer = 1 To n
            Console.WriteLine(2 * i)
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 189: Contar cuántos caracteres de una cadena son vocales

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim vocales As String = "aeiouAEIOU"
        Dim contador As Integer = 0

        For Each c As Char In texto
            If vocales.Contains(c) Then contador += 1
        Next

        Console.WriteLine($"Cantidad de vocales: {contador}")
        Console.ReadLine()
    End Sub
```

```
End Module
```

Ejercicio 190: Sumar dos matrices 3x3

```
Module Module1
    Sub Main()
        Dim matriz1(2, 2) As Integer
        Dim matriz2(2, 2) As Integer
        Dim suma(2, 2) As Integer

        Console.WriteLine("Ingrese los elementos de la primera matriz (3x3):")
        For i As Integer = 0 To 2
            For j As Integer = 0 To 2
                Console.Write($"Elemento [{i},{j}]: ")
                matriz1(i, j) = Convert.ToInt32(Console.ReadLine())
            Next
        Next

        Console.WriteLine("Ingrese los elementos de la segunda matriz (3x3):")
        For i As Integer = 0 To 2
            For j As Integer = 0 To 2
                Console.Write($"Elemento [{i},{j}]: ")
                matriz2(i, j) = Convert.ToInt32(Console.ReadLine())
            Next
        Next

        For i As Integer = 0 To 2
            For j As Integer = 0 To 2
                suma(i, j) = matriz1(i, j) + matriz2(i, j)
            Next
        Next

        Console.WriteLine("Matriz suma:")
        For i As Integer = 0 To 2
            For j As Integer = 0 To 2
                Console.Write(suma(i, j) & vbTab)
            Next
            Console.WriteLine()
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 191: Calcular la potencia de un número usando recursión

```
Module Module1
    Function Potencia(base As Double, exponente As Integer) As Double
        If exponente = 0 Then
            Return 1
        ElseIf exponente > 0 Then
            Return base * Potencia(base, exponente - 1)
        Else
            Return 1 / Potencia(base, -exponente)
        End If
    End Function

    Sub Main()
```

```
        Console.Write("Ingrese la base: ")
        Dim b As Double = Convert.ToDouble(Console.ReadLine())
        Console.Write("Ingrese el exponente (entero): ")
        Dim e As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"{b} elevado a {e} es: {Potencia(b, e)}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 192: Contar la cantidad de caracteres únicos en una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Dim caracteresUnicos = texto.Distinct().Count()
        Console.WriteLine($"Cantidad de caracteres únicos: {caracteresUnicos}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 193: Generar números primos hasta un límite N

```
Module Module1
    Function EsPrimo(num As Integer) As Boolean
        If num < 2 Then Return False
        For i As Integer = 2 To Math.Sqrt(num)
            If num Mod i = 0 Then Return False
        Next
        Return True
    End Function

    Sub Main()
        Console.Write("Ingrese el límite: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Console.WriteLine($"Números primos hasta {n}:")
        For i As Integer = 2 To n
            If EsPrimo(i) Then Console.Write(i & " ")
        Next
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 194: Mostrar los dígitos de un número en orden inverso (sin convertir a cadena)

```
Module Module1
    Sub Main()
        Console.Write("Ingrese un número entero: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Console.Write("Dígitos invertidos: ")

        Dim n As Integer = Math.Abs(num)
        While n > 0
            Console.Write(n Mod 10 & " ")
```

```
            n \= 10
        End While

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 195: Ordenar un arreglo de cadenas alfabéticamente

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de palabras: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim palabras(n - 1) As String

        For i As Integer = 0 To n - 1
            Console.Write($"Palabra {i + 1}: ")
            palabras(i) = Console.ReadLine()
        Next

        Array.Sort(palabras)
        Console.WriteLine("Palabras ordenadas alfabéticamente:")
        For Each palabra In palabras
            Console.WriteLine(palabra)
        Next

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 196: Encontrar el índice de la primera aparición de un carácter en una cadena

```
Module Module1
    Sub Main()
        Console.Write("Ingrese una cadena: ")
        Dim texto As String = Console.ReadLine()
        Console.Write("Ingrese un carácter a buscar: ")
        Dim c As Char = Console.ReadLine()(0)

        Dim indice As Integer = texto.IndexOf(c)
        If indice >= 0 Then
            Console.WriteLine($"El carácter '{c}' aparece primero en la posición
{indice}.")
        Else
            Console.WriteLine($"El carácter '{c}' no se encuentra en la
cadena.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 197: Calcular el promedio de notas y determinar si aprobó (nota ≥ 6)

```
Module Module1
    Sub Main()
        Console.Write("Ingrese la cantidad de notas: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim suma As Double = 0

        For i As Integer = 1 To n
            Console.Write($"Nota {i}: ")
            Dim nota As Double = Convert.ToDouble(Console.ReadLine())
            suma += nota
        Next

        Dim promedio As Double = suma / n
        Console.WriteLine($"Promedio: {promedio:F2}")

        If promedio >= 6 Then
            Console.WriteLine("Aprobó.")
        Else
            Console.WriteLine("No aprobó.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 198: Validar si un número es capicúa (igual leído al derecho y al revés)

```
Module Module1
    Function EsCapicua(num As Integer) As Boolean
        Dim original As String = num.ToString()
        Dim reverso As String = New String(original.Reverse().ToArray())
        Return original = reverso
    End Function

    Sub Main()
        Console.Write("Ingrese un número: ")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If EsCapicua(num) Then
            Console.WriteLine($"{num} es capicúa.")
        Else
            Console.WriteLine($"{num} no es capicúa.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 199: Sumar los elementos de la diagonal principal de una matriz cuadrada

```
Module Module1
    Sub Main()
        Console.Write("Ingrese el tamaño de la matriz cuadrada: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim matriz(n - 1, n - 1) As Integer
```

```
        For i As Integer = 0 To n - 1
            For j As Integer = 0 To n - 1
                Console.Write($"Elemento [{i},{j}]: ")
                matriz(i, j) = Convert.ToInt32(Console.ReadLine())
            Next
        Next

        Dim suma As Integer = 0
        For i As Integer = 0 To n - 1
            suma += matriz(i, i)
        Next

        Console.WriteLine($"Suma de la diagonal principal: {suma}")
        Console.ReadLine()
    End Sub
End Module
```

Ejercicio 200: Contar cuántos números negativos hay en un arreglo

```
Module Module1
    Sub Main()
        Console.Write("Ingrese tamaño del arreglo: ")
        Dim n As Integer = Convert.ToInt32(Console.ReadLine())
        Dim arr(n - 1) As Integer

        For i As Integer = 0 To n - 1
            Console.Write($"Elemento {i + 1}: ")
            arr(i) = Convert.ToInt32(Console.ReadLine())
        Next

        Dim negativos As Integer = arr.Count(Function(x) x < 0)
        Console.WriteLine($"Cantidad de números negativos: {negativos}")
        Console.ReadLine()
    End Sub
End Module
```

---

200 Ejercicios Visual Basic net de Form

Ejercicio 1: Cambiar el texto del formulario

Enunciado:
Crea un formulario que, al hacer clic en un botón, cambie su título a "¡Bienvenido!".

Solución y explicación:

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Me.Text = "¡Bienvenido!"
    End Sub
End Class
```

Explicación:
El evento `Click` del botón cambia la propiedad `Text` del formulario (`Me` hace referencia al formulario actual).

Ejercicio 2: Mostrar un mensaje al hacer clic en un botón

Enunciado:
Agrega un botón que, al pulsarlo, muestre un mensaje que diga “Hola, usuario”.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    MessageBox.Show("Hola, usuario")
End Sub
```

Explicación:
Se utiliza `MessageBox.Show` para mostrar una ventana emergente con el mensaje.

Ejercicio 3: Cerrar el formulario

Enunciado:
Agrega un botón que cierre el formulario actual al hacer clic.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Close()
End Sub
```

Explicación:
`Me.Close()` cierra la ventana actual del formulario.

Ejercicio 4: Cambiar el color de fondo del formulario

Enunciado:
Agrega un botón que, al hacer clic, cambie el fondo del formulario a azul.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.BackColor = Color.Blue
End Sub
```

Explicación:
La propiedad `BackColor` del formulario permite cambiar el color de fondo.

Ejercicio 5: Cambiar el tamaño del formulario

Enunciado:
Agrega un botón que, al hacer clic, cambie el tamaño del formulario a 800x600.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Size = New Size(800, 600)
End Sub
```

Explicación:
`Size` establece las dimensiones del formulario usando el objeto `Size`.

Ejercicio 6: Mostrar la fecha y hora actual

Enunciado:
Agrega un botón que muestre la fecha y hora actual en un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = DateTime.Now.ToString()
End Sub
```

Explicación:
`DateTime.Now.ToString()` convierte la fecha y hora actual en texto y lo asigna a la propiedad `Text` del `Label`.

Ejercicio 7: Habilitar y deshabilitar un botón

Enunciado:
Agrega dos botones: uno que desactive el otro y uno que lo reactive.

Solución:

```
Private Sub ButtonDesactivar_Click(sender As Object, e As EventArgs) Handles
ButtonDesactivar.Click
    ButtonObjetivo.Enabled = False
End Sub

Private Sub ButtonActivar_Click(sender As Object, e As EventArgs) Handles
ButtonActivar.Click
    ButtonObjetivo.Enabled = True
End Sub
```

Explicación:
La propiedad `Enabled` permite habilitar o deshabilitar botones.

Ejercicio 8: Cambiar el texto de un Label desde un TextBox

Enunciado:
Permite al usuario escribir en un `TextBox` y que al pulsar un botón, el texto aparezca en un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = TextBox1.Text
End Sub
```

Explicación:
`TextBox.Text` obtiene lo que el usuario escribió, y se asigna a `Label.Text`.

Ejercicio 9: Mostrar un saludo personalizado

Enunciado:
Permite que el usuario escriba su nombre en un `TextBox` y al hacer clic en un botón, aparezca "Hola, [nombre]".

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim nombre As String = TextBox1.Text
    MessageBox.Show("Hola, " & nombre)
End Sub
```

Explicación:
Se guarda el contenido del `TextBox` en una variable y se concatena con el saludo.

Ejercicio 10: Cambiar el color de texto de un Label

Enunciado:
Agrega un botón que cambie el color del texto de un `Label` a rojo.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.ForeColor = Color.Red
End Sub
```

Explicación:
`ForeColor` controla el color del texto del `Label`.

Ejercicio 11: Limpiar un TextBox

Enunciado:
Agrega un botón que al hacer clic limpie el contenido de un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    TextBox1.Clear()
End Sub
```

Explicación:
El método `Clear()` borra todo el texto dentro del `TextBox`.

Ejercicio 12: Mostrar texto en mayúsculas

Enunciado:
Al hacer clic en un botón, muestra en un `Label` el texto del `TextBox` convertido a mayúsculas.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = TextBox1.Text.ToUpper()
End Sub
```

Explicación:
El método `ToUpper()` convierte una cadena a mayúsculas.

Ejercicio 13: Contar caracteres de un TextBox

Enunciado:
Al hacer clic en un botón, muestra la cantidad de caracteres escritos en el `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Caracteres: " & TextBox1.Text.Length
End Sub
```

Explicación:
La propiedad `Length` indica cuántos caracteres tiene el texto.

Ejercicio 14: Agregar elementos a un ListBox

Enunciado:
Permite ingresar un texto en un `TextBox` y al pulsar un botón, añadirlo a un `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    ListBox1.Items.Add(TextBox1.Text)
End Sub
```

Explicación:
`ListBox.Items.Add` añade un nuevo ítem a la lista.

Ejercicio 15: Eliminar elemento seleccionado en ListBox

Enunciado:
Agrega un botón que elimine el ítem seleccionado en el `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    ListBox1.Items.Remove(ListBox1.SelectedItem)
End Sub
```

Explicación:
`SelectedItem` obtiene el elemento seleccionado y `Remove` lo elimina.

Ejercicio 16: Mostrar elemento seleccionado de ListBox en Label

Enunciado:
Cuando se selecciona un ítem del `ListBox`, mostrarlo en un `Label`.

Solución:

```
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ListBox1.SelectedIndexChanged
    Label1.Text = "Seleccionado: " & ListBox1.SelectedItem
End Sub
```

Explicación:
Se utiliza el evento `SelectedIndexChanged` y se muestra el ítem actual en el `Label`.

Ejercicio 17: Usar un ComboBox para seleccionar colores

Enunciado:
Agrega un `ComboBox` con colores. Al seleccionar uno, cambia el fondo del formulario a ese color.

Solución:

```
Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ComboBox1.SelectedIndexChanged
    Me.BackColor = Color.FromName(ComboBox1.SelectedItem)
End Sub
```

Explicación:
`Color.FromName` convierte el texto del `ComboBox` en un color válido.

Ejercicio 18: Calcular la suma de dos números

Enunciado:
Permite al usuario ingresar dos números en dos `TextBox` y mostrar la suma al hacer clic.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim a As Integer = Val(TextBox1.Text)
    Dim b As Integer = Val(TextBox2.Text)
    Label1.Text = "Suma: " & (a + b)
End Sub
```

Explicación:
`Val()` convierte texto a número. Se suman y se muestran en el `Label`.

Ejercicio 19: Usar un CheckBox para activar/desactivar un botón

Enunciado:
Agrega un `CheckBox`. Si está marcado, activa un botón; si no, lo desactiva.

Solución:

```
Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles
CheckBox1.CheckedChanged
    Button1.Enabled = CheckBox1.Checked
End Sub
```

Explicación:
`Checked` es `True` cuando está marcado, y se usa para controlar `Enabled`.

Ejercicio 20: Mostrar contraseña oculta en TextBox

Enunciado:
Agrega un `CheckBox` que permita ver/ocultar el contenido de un `TextBox` con contraseña.

Solución:

```
Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles
CheckBox1.CheckedChanged
    If CheckBox1.Checked Then
        TextBox1.PasswordChar = ""
    Else
        TextBox1.PasswordChar = "*"
    End If
End Sub
```

Explicación:
`PasswordChar` establece el carácter que oculta la contraseña. Vacío muestra el texto.

Ejercicio 21: Usar un ProgressBar con un temporizador

Enunciado:
Agrega un `ProgressBar` y un `Timer`. Cada vez que el timer se dispare, aumenta el valor del `ProgressBar`.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value < ProgressBar1.Maximum Then
        ProgressBar1.Value += 1
    Else
        Timer1.Stop()
    End If
End Sub
```

Explicación:
El `Timer` incrementa el `ProgressBar` hasta su valor máximo, luego se detiene.

Ejercicio 22: Reiniciar ProgressBar

Enunciado:
Agrega un botón que reinicie el `ProgressBar` a cero.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    ProgressBar1.Value = 0
End Sub
```

Explicación:
La propiedad `Value` controla el valor actual de la barra de progreso.

Ejercicio 23: Cambiar el tamaño de fuente de un Label

Enunciado:
Agrega botones para aumentar y disminuir el tamaño del texto de un `Label`.

Solución:

```
Private Sub ButtonAumentar_Click(sender As Object, e As EventArgs) Handles
ButtonAumentar.Click
    Label1.Font = New Font(Label1.Font.FontFamily, Label1.Font.Size + 2)
End Sub

Private Sub ButtonDisminuir_Click(sender As Object, e As EventArgs) Handles
ButtonDisminuir.Click
    Label1.Font = New Font(Label1.Font.FontFamily, Label1.Font.Size - 2)
End Sub
```

Explicación:
Se crea un nuevo objeto `Font` con el mismo tipo de letra pero diferente tamaño.

Ejercicio 24: Mostrar fecha con DateTimePicker

Enunciado:
Muestra la fecha seleccionada de un `DateTimePicker` en un `Label`.

Solución:

```
Private Sub DateTimePicker1_ValueChanged(sender As Object, e As EventArgs)
Handles DateTimePicker1.ValueChanged
    Label1.Text = DateTimePicker1.Value.ToShortDateString()
End Sub
```

Explicación:
`Value` devuelve la fecha seleccionada. `ToShortDateString` la convierte en formato corto.

Ejercicio 25: Confirmar antes de cerrar el formulario

Enunciado:
Antes de cerrar el formulario, preguntar al usuario si está seguro.

Solución:

```
Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs)
Handles MyBase.FormClosing
    If MessageBox.Show("¿Seguro que quieres salir?", "Confirmar",
MessageBoxButtons.YesNo) = DialogResult.No Then
        e.Cancel = True
    End If
End Sub
```

Explicación:
El evento `FormClosing` permite cancelar el cierre si el usuario responde "No".

Ejercicio 26: Mostrar número de clicks en botón

Enunciado:
Cada vez que se pulse un botón, contar cuántas veces se ha hecho clic.

Solución:

```
Dim contador As Integer = 0

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    contador += 1
    Label1.Text = "Clicks: " & contador
End Sub
```

Explicación:
Una variable cuenta clics y se actualiza el `Label` con cada evento.

Ejercicio 27: Cambiar imagen de un PictureBox

Enunciado:
Agrega un botón que cambie la imagen mostrada en un `PictureBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    PictureBox1.Image = Image.FromFile("C:\Ruta\imagen.jpg")
End Sub
```

Explicación:
`Image.FromFile` carga una imagen desde disco. Asegúrate de que la ruta sea válida.

Ejercicio 28: Reproducir un sonido WAV

Enunciado:
Agrega un botón que reproduzca un sonido `.wav`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    My.Computer.Audio.Play("C:\Ruta\sonido.wav")
End Sub
```

Explicación:
`My.Computer.Audio.Play` permite reproducir sonidos en VB.NET.

Ejercicio 29: Usar RadioButtons para elegir un color

Enunciado:
Agrega tres `RadioButtons` para elegir entre rojo, verde o azul, y aplicar el color a un `Label`.

Solución:

```
Private Sub RadioButton_CheckedChanged(sender As Object, e As EventArgs) Handles
RadioButton1.CheckedChanged, RadioButton2.CheckedChanged,
RadioButton3.CheckedChanged
```

```
    If RadioButton1.Checked Then
        Label1.ForeColor = Color.Red
    ElseIf RadioButton2.Checked Then
        Label1.ForeColor = Color.Green
    ElseIf RadioButton3.Checked Then
        Label1.ForeColor = Color.Blue
    End If
End Sub
```

Explicación:
Cada `RadioButton` cambia el color del texto del `Label`.

Ejercicio 30: Mostrar hora en tiempo real con Timer

Enunciado:
Usa un `Timer` para actualizar cada segundo un `Label` con la hora actual.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = DateTime.Now.ToLongTimeString()
End Sub
```

Explicación:
`DateTime.Now.ToLongTimeString()` da la hora actual en formato largo. El `Timer` debe tener `Interval = 1000`.

Ejercicio 31: Bloquear el cierre del formulario si un campo está vacío

Enunciado:
Evita que se cierre el formulario si un `TextBox` está vacío.

Solución:

```
Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs)
Handles MyBase.FormClosing
    If TextBox1.Text.Trim() = "" Then
        MessageBox.Show("El campo no puede estar vacío.")
        e.Cancel = True
    End If
End Sub
```

Explicación:
Si el texto está vacío, se muestra un mensaje y se cancela el cierre del formulario.

Ejercicio 32: Centrar el formulario al iniciar

Enunciado:
Haz que el formulario se muestre centrado en la pantalla al ejecutarse.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Me.StartPosition = FormStartPosition.CenterScreen
End Sub
```

Explicación:
La propiedad `StartPosition` controla la posición inicial del formulario.

Ejercicio 33: Abrir un formulario secundario desde uno principal

Enunciado:
Agrega un botón que abra otro formulario llamado `Form2`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim nuevoFormulario As New Form2()
    nuevoFormulario.Show()
End Sub
```

Explicación:
Se instancia `Form2` y se muestra con `Show()`.

Ejercicio 34: Abrir un formulario como diálogo (modal)

Enunciado:
Abre `Form2` como ventana modal desde `Form1`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim frm As New Form2()
    frm.ShowDialog()
End Sub
```

Explicación:
`ShowDialog()` abre el formulario como ventana modal, bloqueando la interacción con el resto.

Ejercicio 35: Ocultar el formulario actual

Enunciado:
Al hacer clic en un botón, oculta el formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Hide()
End Sub
```

Explicación:
`Me.Hide()` oculta el formulario actual sin cerrarlo.

Ejercicio 36: Cerrar un formulario con botón

Enunciado:
Agrega un botón que cierre el formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Close()
End Sub
```

Explicación:
`Me.Close()` cierra el formulario activo.

Ejercicio 37: Mostrar número aleatorio en un Label

Enunciado:
Genera un número aleatorio entre 1 y 100 al pulsar un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim rnd As New Random()
    Label1.Text = rnd.Next(1, 101).ToString()
End Sub
```

Explicación:
`Random.Next(1, 101)` genera números del 1 al 100.

Ejercicio 38: Mostrar mensaje con el nombre de usuario ingresado

Enunciado:
Al pulsar un botón, muestra un saludo con el nombre escrito en un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    MessageBox.Show("Hola, " & TextBox1.Text & "!")
End Sub
```

Explicación:
`MessageBox.Show` muestra una ventana emergente con el texto concatenado.

Ejercicio 39: Evitar que se ingresen letras en un TextBox numérico

Enunciado:
Evita que el usuario escriba letras en un campo solo numérico.

Solución:

```
Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox1.KeyPress
    If Not Char.IsDigit(e.KeyChar) AndAlso Not Char.IsControl(e.KeyChar) Then
        e.Handled = True
    End If
End Sub
```

Explicación:
`IsDigit` permite solo números, `IsControl` deja usar teclas como borrar.

Ejercicio 40: Hacer un contador con botones (+1 y -1)

Enunciado:
Crea dos botones: uno para sumar 1 al contador y otro para restar 1.

Solución:

```
Dim contador As Integer = 0

Private Sub ButtonSuma_Click(sender As Object, e As EventArgs) Handles
ButtonSuma.Click
    contador += 1
    Label1.Text = contador.ToString()
End Sub

Private Sub ButtonResta_Click(sender As Object, e As EventArgs) Handles
ButtonResta.Click
    contador -= 1
    Label1.Text = contador.ToString()
End Sub
```

Explicación:
Se actualiza una variable global `contador` con los clics y se muestra en el `Label`.

Ejercicio 41: Cambiar el título del formulario desde un TextBox

Enunciado:
Escribe un texto en un `TextBox` y cambia el título del formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Text = TextBox1.Text
End Sub
```

Explicación:
La propiedad `Text` del formulario controla su título.

Ejercicio 42: Hacer visible un control oculto

Enunciado:
Un `Label` está oculto. Hazlo visible al pulsar un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Visible = True
End Sub
```

Explicación:
`Visible` determina si un control está o no visible en el formulario.

Ejercicio 43: Crear un temporizador para cuenta regresiva

Enunciado:
Usa un `Timer` para hacer una cuenta atrás de 10 a 0 en un `Label`.

Solución:

```
Dim tiempo As Integer = 10

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = tiempo.ToString()
    tiempo -= 1
    If tiempo < 0 Then
        Timer1.Stop()
        MessageBox.Show("¡Fin del tiempo!")
    End If
End Sub
```

Explicación:
Cada tick del `Timer` resta 1 y detiene el conteo al llegar a 0.

Ejercicio 44: Seleccionar archivo con OpenFileDialog

Enunciado:
Abre un cuadro para seleccionar un archivo y muestra su ruta.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim dlg As New OpenFileDialog()
    If dlg.ShowDialog() = DialogResult.OK Then
        Label1.Text = dlg.FileName
    End If
End Sub
```

Explicación:
`OpenFileDialog` permite elegir archivos. Se muestra la ruta seleccionada.

Ejercicio 45: Cambiar el color de fondo con ColorDialog

Enunciado:
Abre un selector de color y cambia el fondo del formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim dlg As New ColorDialog()
    If dlg.ShowDialog() = DialogResult.OK Then
        Me.BackColor = dlg.Color
    End If
End Sub
```

Explicación:
`ColorDialog` permite elegir un color y se aplica al fondo del formulario.

Ejercicio 46: Agregar elementos por teclado al ListBox (Enter)

Enunciado:
Permite añadir texto al `ListBox` al pulsar Enter en el `TextBox`.

Solución:

```
Private Sub TextBox1_KeyDown(sender As Object, e As KeyEventArgs) Handles
TextBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        ListBox1.Items.Add(TextBox1.Text)
        TextBox1.Clear()
    End If
End Sub
```

Explicación:
Detecta la tecla `Enter` y agrega el texto a la lista.

Ejercicio 47: Contar elementos de un ListBox

Enunciado:
Al pulsar un botón, muestra cuántos elementos hay en un `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Total: " & ListBox1.Items.Count
End Sub
```

Explicación:
`Items.Count` devuelve el número de elementos en el `ListBox`.

Ejercicio 48: Validar que un campo esté lleno antes de continuar

Enunciado:
Si el `TextBox` está vacío, mostrar un error y no continuar.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If TextBox1.Text.Trim() = "" Then
        MessageBox.Show("El campo no puede estar vacío.")
    Else
        MessageBox.Show("Campo válido.")
    End If
End Sub
```

Explicación:
Se verifica si el contenido del campo está vacío usando `Trim()`.

Ejercicio 49: Hacer que un botón parpadee (cambiar color)

Enunciado:
Con un `Timer`, cambia el color del botón cada segundo.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If Button1.BackColor = Color.Red Then
        Button1.BackColor = Color.Blue
    Else
        Button1.BackColor = Color.Red
    End If
End Sub
```

Explicación:
Alterna el color entre rojo y azul cada vez que el `Timer` se activa.

Ejercicio 50: Cargar texto desde archivo a TextBox

Enunciado:
Abre un archivo de texto y muestra su contenido en un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim dlg As New OpenFileDialog()
    If dlg.ShowDialog() = DialogResult.OK Then
        TextBox1.Text = IO.File.ReadAllText(dlg.FileName)
    End If
```

```
End Sub
```

Explicación:
`ReadAllText` lee el contenido completo de un archivo y lo muestra.

Ejercicio 51: Guardar texto de un TextBox en un archivo

Enunciado:
Guarda el contenido de un `TextBox` en un archivo de texto usando `SaveFileDialog`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim dlg As New SaveFileDialog()
    dlg.Filter = "Archivos de texto|*.txt"
    If dlg.ShowDialog() = DialogResult.OK Then
        IO.File.WriteAllText(dlg.FileName, TextBox1.Text)
    End If
End Sub
```

Explicación:
Se usa `SaveFileDialog` para pedir al usuario la ruta de guardado, y luego `WriteAllText` guarda el contenido.

Ejercicio 52: Maximizar el formulario al iniciar

Enunciado:
Haz que el formulario se abra en modo maximizado.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Me.WindowState = FormWindowState.Maximized
End Sub
```

Explicación:
La propiedad `WindowState` controla si el formulario se abre normal, minimizado o maximizado.

Ejercicio 53: Desactivar un botón después de pulsarlo

Enunciado:
Desactiva un botón tras hacer clic en él.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Button1.Enabled = False
End Sub
```

Explicación:
`Enabled = False` inhabilita el botón para que no se pueda volver a pulsar.

Ejercicio 54: Contar caracteres de un TextBox en tiempo real

Enunciado:
Muestra en un `Label` el número de caracteres mientras se escribe.

Solución:

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles
TextBox1.TextChanged
    Label1.Text = "Caracteres: " & TextBox1.Text.Length
End Sub
```

Explicación:
`TextChanged` se activa al escribir, y `Length` cuenta los caracteres del texto.

Ejercicio 55: Mostrar un reloj en tiempo real

Enunciado:
Muestra la hora actual en un `Label` usando un `Timer`.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = DateTime.Now.ToString("HH:mm:ss")
End Sub
```

Explicación:
`DateTime.Now` da la hora actual, y el `Timer` actualiza el `Label` cada segundo.

Ejercicio 56: Cambiar color de fondo del formulario al azar

Enunciado:
Al hacer clic en un botón, cambia el fondo a un color aleatorio.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim rnd As New Random()
    Dim color As Color = Color.FromArgb(rnd.Next(256), rnd.Next(256),
rnd.Next(256))
    Me.BackColor = color
End Sub
```

Explicación:
`FromArgb` crea un color a partir de componentes RGB aleatorios.

Ejercicio 57: Mostrar mensaje de despedida al cerrar

Enunciado:
Al cerrar el formulario, muestra "Gracias por usar la aplicación".

Solución:

```
Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs)
Handles MyBase.FormClosing
    MessageBox.Show("Gracias por usar la aplicación.")
End Sub
```

Explicación:
`FormClosing` se ejecuta justo antes de que el formulario se cierre.

Ejercicio 58: Seleccionar todos los elementos de un ListBox

Enunciado:
Agrega un botón que seleccione todos los elementos del `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    For i As Integer = 0 To ListBox1.Items.Count - 1
        ListBox1.SetSelected(i, True)
    Next
End Sub
```

Explicación:
`SetSelected(i, True)` marca como seleccionado el elemento en la posición `i`.

Ejercicio 59: Mostrar el índice del elemento seleccionado en un ListBox

Enunciado:
Muestra el índice del ítem seleccionado en un `Label`.

Solución:

```
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ListBox1.SelectedIndexChanged
    Label1.Text = "Índice: " & ListBox1.SelectedIndex
End Sub
```

Explicación:
`SelectedIndex` devuelve la posición del elemento elegido.

Ejercicio 60: Validar formato de correo electrónico

Enunciado:
Verifica si el texto del `TextBox` tiene un formato de correo válido.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim email As String = TextBox1.Text
    If email.Contains("@") AndAlso email.Contains(".") Then
        MessageBox.Show("Correo válido.")
    Else
        MessageBox.Show("Formato incorrecto.")
    End If
End Sub
```

Explicación:
Se verifica de forma básica si contiene `@` y `.` como validación mínima.

Ejercicio 61: Mostrar el día de la semana actual

Enunciado:
Muestra el día actual (lunes, martes, etc.) en un `Label`.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Label1.Text = "Hoy es " & DateTime.Now.ToString("dddd")
End Sub
```

Explicación:
El formato `"dddd"` extrae el nombre del día de la semana.

Ejercicio 62: Sumar dos números ingresados en dos TextBox

Enunciado:
Ingresa dos números y muestra su suma.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim a As Integer = Val(TextBox1.Text)
    Dim b As Integer = Val(TextBox2.Text)
    Label1.Text = "Suma: " & (a + b).ToString()
End Sub
```

Explicación:
`Val` convierte el texto a número, y se muestra la suma en el `Label`.

Ejercicio 63: Mostrar el nombre del mes actual

Enunciado:
Muestra el mes actual (enero, febrero, etc.) en un `Label`.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Label1.Text = "Mes actual: " & DateTime.Now.ToString("MMMM")
End Sub
```

Explicación:
El formato `"MMMM"` devuelve el nombre completo del mes.

Ejercicio 64: Hacer un TextBox de solo lectura

Enunciado:
Haz que el `TextBox1` no permita que el usuario escriba.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    TextBox1.ReadOnly = True
End Sub
```

Explicación:
La propiedad `ReadOnly` evita edición sin ocultar el texto.

Ejercicio 65: Mostrar una barra de progreso creciente

Enunciado:
Usa un `Timer` para aumentar el valor de una `ProgressBar`.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value < ProgressBar1.Maximum Then
        ProgressBar1.Value += 1
    Else
        Timer1.Stop()
        MessageBox.Show("Completado.")
    End If
End Sub
```

Explicación:
Cada tick aumenta el valor hasta el máximo y se detiene.

Ejercicio 66: Mostrar mensaje personalizado según la hora

Enunciado:
Mostrar "Buenos días", "Buenas tardes" o "Buenas noches" según la hora.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Dim hora As Integer = DateTime.Now.Hour
    If hora < 12 Then
        Label1.Text = "Buenos días"
    ElseIf hora < 20 Then
        Label1.Text = "Buenas tardes"
```

```
    Else
        Label1.Text = "Buenas noches"
    End If
End Sub
```

Explicación:
Se evalúa la hora y se muestra un mensaje según el rango horario.

Ejercicio 67: Contar cuántas veces se ha pulsado un botón

Enunciado:
Cuenta y muestra cuántas veces se ha pulsado un botón.

Solución:

```
Dim contador As Integer = 0

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    contador += 1
    Label1.Text = "Pulsado: " & contador
End Sub
```

Explicación:
Una variable incrementa en cada clic y se muestra en pantalla.

Ejercicio 68: Mostrar un mensaje solo una vez

Enunciado:
Al pulsar un botón, se muestra un mensaje solo la primera vez.

Solución:

```
Dim mostrado As Boolean = False

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If Not mostrado Then
        MessageBox.Show("Este mensaje solo aparece una vez.")
        mostrado = True
    End If
End Sub
```

Explicación:
Una variable controla si el mensaje ya ha sido mostrado.

Ejercicio 69: Desplazar un Label a la derecha con un botón

Enunciado:
Cada vez que pulses un botón, mueve un `Label` 10 píxeles a la derecha.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Left += 10
End Sub
```

Explicación:
`Left` controla la posición horizontal del control.

Ejercicio 70: Cambiar el texto de un botón al hacer clic

Enunciado:
Haz que el texto del botón cambie a "¡Clic hecho!" después de pulsarlo.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Button1.Text = "¡Clic hecho!"
End Sub
```

Explicación:
`Text` permite modificar el texto que aparece en el botón.

Ejercicio 71: Alternar visibilidad de un Label

Enunciado:
Cada vez que pulses un botón, alterna la visibilidad de un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Visible = Not Label1.Visible
End Sub
```

Explicación:
La propiedad `Visible` permite ocultar o mostrar el `Label`. Usamos `Not` para invertir su valor.

Ejercicio 72: Limpiar todos los TextBox del formulario

Enunciado:
Crea un botón que borre el contenido de todos los `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    For Each ctrl As Control In Me.Controls
        If TypeOf ctrl Is TextBox Then
            CType(ctrl, TextBox).Clear()
        End If
    Next
End Sub
```

Explicación:
Recorremos todos los controles del formulario y limpiamos los que sean `TextBox`.

Ejercicio 73: Deshabilitar todos los botones del formulario

Enunciado:
Agrega un botón que deshabilite todos los botones del formulario, incluido él mismo.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    For Each ctrl As Control In Me.Controls
        If TypeOf ctrl Is Button Then
            ctrl.Enabled = False
        End If
    Next
End Sub
```

Explicación:
Recorre los controles y desactiva todos los que sean botones.

Ejercicio 74: Cambiar la fuente de un Label

Enunciado:
Al hacer clic en un botón, cambia la fuente y el tamaño del texto de un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Font = New Font("Arial", 16, FontStyle.Bold)
End Sub
```

Explicación:
La propiedad `Font` define el tipo, tamaño y estilo del texto.

Ejercicio 75: Hacer parpadear un Label

Enunciado:
Con un `Timer`, haz que un `Label` parpadee (aparezca y desaparezca).

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Visible = Not Label1.Visible
End Sub
```

Explicación:
Alterna la visibilidad del `Label` en cada tick del `Timer`.

Ejercicio 76: Cerrar el formulario desde un botón

Enunciado:
Al pulsar un botón, cierra la aplicación.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Close()
End Sub
```

Explicación:
`Me.Close()` cierra el formulario actual.

Ejercicio 77: Redondear un número decimal de un TextBox

Enunciado:
Al pulsar un botón, redondea el número ingresado y lo muestra en un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim numero As Double = Val(TextBox1.Text)
    Label1.Text = "Redondeado: " & Math.Round(numero)
End Sub
```

Explicación:
Usamos `Math.Round` para redondear un número decimal.

Ejercicio 78: Mostrar fecha actual con formato largo

Enunciado:
Muestra la fecha actual en formato largo (por ejemplo: "jueves, 27 de junio de 2025").

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Label1.Text = DateTime.Now.ToLongDateString()
End Sub
```

Explicación:
`ToLongDateString()` devuelve la fecha en formato completo.

Ejercicio 79: Contar cuántos CheckBox están marcados

Enunciado:
Muestra cuántos `CheckBox` están marcados en el formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim contador As Integer = 0
    For Each ctrl As Control In Me.Controls
        If TypeOf ctrl Is CheckBox AndAlso CType(ctrl, CheckBox).Checked Then
            contador += 1
        End If
    Next
    Label1.Text = "Marcados: " & contador
End Sub
```

Explicación:
Se recorre cada `CheckBox` y se cuenta si está marcado (`Checked = True`).

Ejercicio 80: Mostrar nombre de usuario ingresado

Enunciado:
Pide al usuario su nombre en un `TextBox` y lo saluda en un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Hola, " & TextBox1.Text & "!"
End Sub
```

Explicación:
Concatena el texto del `TextBox` en el mensaje de saludo.

Ejercicio 81: Crear una calculadora de resta

Enunciado:
Permite ingresar dos números y muestra su resta.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim a As Double = Val(TextBox1.Text)
    Dim b As Double = Val(TextBox2.Text)
    Label1.Text = "Resultado: " & (a - b)
End Sub
```

Explicación:
Simple operación de resta con entrada desde dos `TextBox`.

Ejercicio 82: Establecer título del formulario con un TextBox

Enunciado:
Permite al usuario cambiar el título del formulario escribiéndolo en un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Text = TextBox1.Text
End Sub
```

Explicación:
La propiedad `Text` del formulario define su título.

Ejercicio 83: Mostrar un mensaje si un CheckBox está marcado

Enunciado:
Al hacer clic en un botón, si un `CheckBox` está marcado, mostrar "Aceptado".

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If CheckBox1.Checked Then
        MessageBox.Show("Aceptado")
    Else
        MessageBox.Show("No aceptado")
    End If
End Sub
```

Explicación:
Se evalúa si `Checked` es `True` o `False`.

Ejercicio 84: Crear un contador regresivo de 10 a 0

Enunciado:
Con un `Timer`, muestra un contador regresivo en un `Label`.

Solución:

```
Dim contador As Integer = 10

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = contador
    contador -= 1
    If contador < 0 Then Timer1.Stop()
End Sub
```

Explicación:
Decrementa el contador cada segundo y se detiene en 0.

Ejercicio 85: Mostrar el número más grande entre dos TextBox

Enunciado:
Compara dos números y muestra cuál es el mayor.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim a As Double = Val(TextBox1.Text)
    Dim b As Double = Val(TextBox2.Text)
    Label1.Text = "Mayor: " & Math.Max(a, b)
End Sub
```

Explicación:
`Math.Max` devuelve el número mayor entre dos valores.

Ejercicio 86: Crear una función que devuelva el doble de un número

Enunciado:
Crea una función que reciba un número y devuelva el doble. Muestra el resultado al pulsar un botón.

Solución:

```
Function Doble(valor As Integer) As Integer
    Return valor * 2
End Function

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim numero As Integer = Val(TextBox1.Text)
    Label1.Text = "Doble: " & Doble(numero)
End Sub
```

Explicación:
La función `Doble` multiplica por 2 y se llama desde el botón.

Ejercicio 87: Mostrar si un número es par o impar

Enunciado:
Ingresa un número y muestra si es par o impar.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim n As Integer = Val(TextBox1.Text)
    If n Mod 2 = 0 Then
        Label1.Text = "Par"
    Else
        Label1.Text = "Impar"
    End If
End Sub
```

Explicación:
Usa el operador `Mod` para obtener el resto de dividir por 2.

Ejercicio 88: Crear un botón que aumente el tamaño del formulario

Enunciado:
Al hacer clic, agranda el formulario en 20 píxeles de ancho y alto.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.Width += 20
    Me.Height += 20
End Sub
```

Explicación:
Modifica directamente las dimensiones del formulario.

Ejercicio 89: Mostrar cuántos caracteres tiene el nombre ingresado

Enunciado:
Cuenta cuántos caracteres tiene el texto de un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Longitud: " & TextBox1.Text.Length
End Sub
```

Explicación:
`Text.Length` devuelve el número total de caracteres.

Ejercicio 90: Mostrar nombre completo usando dos TextBox

Enunciado:
Concatena nombre y apellido en un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Nombre completo: " & TextBox1.Text & " " & TextBox2.Text
End Sub
```

Explicación:
Usamos concatenación para unir ambos textos con un espacio entre ellos.

Ejercicio 91: Convertir texto a mayúsculas

Enunciado:
Al hacer clic en un botón, convierte el contenido de un `TextBox` a mayúsculas.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    TextBox1.Text = TextBox1.Text.ToUpper()
End Sub
```

Explicación:
La función `ToUpper` transforma el texto a letras mayúsculas.

Ejercicio 92: Convertir texto a minúsculas

Enunciado:
Transforma el texto de un `TextBox` a minúsculas al pulsar un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    TextBox1.Text = TextBox1.Text.ToLower()
End Sub
```

Explicación:
`ToLower` convierte todas las letras del texto a minúsculas.

Ejercicio 93: Deshabilitar un TextBox desde un botón

Enunciado:
Desactiva un `TextBox` haciendo clic en un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    TextBox1.Enabled = False
End Sub
```

Explicación:
La propiedad `Enabled` controla si el `TextBox` está activo o no.

Ejercicio 94: Contar palabras de un TextBox

Enunciado:
Cuenta cuántas palabras hay en el `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim palabras() As String = TextBox1.Text.Split(" "c)
    Label1.Text = "Palabras: " & palabras.Length
End Sub
```

Explicación:
`Split(" "c)` divide el texto por espacios y devuelve un array de palabras.

Ejercicio 95: Mostrar la hora actual en un Label

Enunciado:
Muestra la hora actual en un `Label` usando un `Timer`.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = DateTime.Now.ToLongTimeString()
End Sub
```

Explicación:
`ToLongTimeString()` da la hora en formato largo (con segundos).

Ejercicio 96: Cambiar el color de fondo del formulario

Enunciado:
Cambia el color de fondo del formulario al pulsar un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.BackColor = Color.LightBlue
End Sub
```

Explicación:
La propiedad `BackColor` permite definir el color de fondo.

Ejercicio 97: Mostrar una lista de países en un ComboBox

Enunciado:
Agrega algunos países a un `ComboBox` en la carga del formulario.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ComboBox1.Items.Add("España")
    ComboBox1.Items.Add("México")
    ComboBox1.Items.Add("Argentina")
End Sub
```

Explicación:
`Items.Add` añade elementos al `ComboBox`.

Ejercicio 98: Mostrar país seleccionado de un ComboBox

Enunciado:
Muestra el país seleccionado en un `Label`.

Solución:

```
Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ComboBox1.SelectedIndexChanged
    Label1.Text = "País: " & ComboBox1.SelectedItem.ToString()
End Sub
```

Explicación:
`SelectedItem` obtiene el ítem elegido por el usuario.

Ejercicio 99: Eliminar un ítem seleccionado de un ListBox

Enunciado:
Elimina el elemento seleccionado del `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If ListBox1.SelectedIndex >= 0 Then
        ListBox1.Items.RemoveAt(ListBox1.SelectedIndex)
    End If
End Sub
```

Explicación:
`RemoveAt` elimina el ítem en el índice especificado.

Ejercicio 100: Calcular promedio de tres notas

Enunciado:
Muestra el promedio de tres notas ingresadas en `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim a = Val(TextBox1.Text)
    Dim b = Val(TextBox2.Text)
    Dim c = Val(TextBox3.Text)
    Label1.Text = "Promedio: " & ((a + b + c) / 3).ToString("0.00")
End Sub
```

Explicación:
Calcula la media y la muestra con dos decimales.

Ejercicio 101: Agregar ítems a un ListBox desde un TextBox

Enunciado:
Al pulsar un botón, añade el contenido del `TextBox` a un `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If TextBox1.Text <> "" Then
        ListBox1.Items.Add(TextBox1.Text)
        TextBox1.Clear()
    End If
End Sub
```

Explicación:
Añade lo ingresado y borra el campo para escribir otro valor.

Ejercicio 102: Mostrar número de elementos en ListBox

Enunciado:
Muestra cuántos elementos hay en un `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Total elementos: " & ListBox1.Items.Count
End Sub
```

Explicación:
`Items.Count` devuelve la cantidad de ítems del `ListBox`.

Ejercicio 103: Limpiar todos los ítems del ListBox

Enunciado:
Borra todos los elementos de un `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    ListBox1.Items.Clear()
End Sub
```

Explicación:
`Clear()` elimina todos los ítems de la lista.

Ejercicio 104: Seleccionar todos los CheckBox del formulario

Enunciado:
Marca todos los `CheckBox` que estén en el formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    For Each ctrl As Control In Me.Controls
        If TypeOf ctrl Is CheckBox Then
            CType(ctrl, CheckBox).Checked = True
        End If
    Next
End Sub
```

Explicación:
Cambia la propiedad `Checked` a `True` para todos los `CheckBox`.

Ejercicio 105: Calcular área de un rectángulo

Enunciado:
Ingresa base y altura y muestra el área del rectángulo.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim b = Val(TextBox1.Text)
    Dim h = Val(TextBox2.Text)
    Label1.Text = "Área: " & (b * h)
End Sub
```

Explicación:
Área = base * altura. El resultado se muestra en un `Label`.

Ejercicio 106: Crear un menú con opciones de color de fondo

Enunciado:
Agrega opciones a un `MenuStrip` para cambiar el color de fondo del formulario.

Solución (fragmento en el evento de cada opción):

```
Private Sub RojoToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles RojoToolStripMenuItem.Click
    Me.BackColor = Color.Red
End Sub
```

Explicación:
Cada opción del menú cambia el color usando `BackColor`.

Ejercicio 107: Mostrar mensaje si el campo está vacío

Enunciado:
Si el `TextBox` está vacío, mostrar advertencia al hacer clic.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If TextBox1.Text = "" Then
        MessageBox.Show("Por favor, escriba algo.")
    End If
End Sub
```

Explicación:
Condicional para comprobar si el campo está vacío (`""`).

Ejercicio 108: Bloquear botón tras pulsarlo una vez

Enunciado:
El botón se desactiva a sí mismo tras el primer clic.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Button1.Enabled = False
End Sub
```

Explicación:
Al desactivar `Enabled`, el botón ya no se puede volver a pulsar.

Ejercicio 109: Confirmar antes de cerrar el formulario

Enunciado:
Solicita confirmación al cerrar el formulario.

Solución:

```
Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs)
Handles MyBase.FormClosing
    If MessageBox.Show("¿Seguro que deseas salir?", "Confirmar",
MessageBoxButtons.YesNo) = DialogResult.No Then
        e.Cancel = True
    End If
End Sub
```

Explicación:
Cancela el cierre si el usuario responde "No".

Ejercicio 110: Abrir un archivo de texto en un TextBox

Enunciado:
Carga el contenido de un archivo `.txt` en un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim openFile As New OpenFileDialog()
```

```
    If openFile.ShowDialog() = DialogResult.OK Then
        TextBox1.Text = IO.File.ReadAllText(openFile.FileName)
    End If
End Sub
```

Explicación:
`OpenFileDialog` permite seleccionar el archivo, luego lo lee con `ReadAllText`.

Ejercicio 111: Validar que un campo contenga solo números

Enunciado:
Muestra un mensaje si el `TextBox` contiene letras.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If Not IsNumeric(TextBox1.Text) Then
        MessageBox.Show("Solo se permiten números.")
    End If
End Sub
```

Explicación:
`IsNumeric` verifica si el contenido puede interpretarse como número.

Ejercicio 112: Contar caracteres de un TextBox

Enunciado:
Muestra la cantidad de caracteres que tiene un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Caracteres: " & TextBox1.Text.Length
End Sub
```

Explicación:
La propiedad `Length` devuelve el número de caracteres del texto.

Ejercicio 113: Mostrar imagen en PictureBox

Enunciado:
Carga una imagen en un `PictureBox` desde un archivo.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim openFile As New OpenFileDialog()
    openFile.Filter = "Imágenes|*.jpg;*.png;*.bmp"
    If openFile.ShowDialog() = DialogResult.OK Then
        PictureBox1.Image = Image.FromFile(openFile.FileName)
```

```
    End If
End Sub
```

Explicación:
`Image.FromFile` carga una imagen desde la ruta seleccionada.

Ejercicio 114: Alternar visibilidad de una imagen

Enunciado:
Muestra u oculta la imagen del `PictureBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    PictureBox1.Visible = Not PictureBox1.Visible
End Sub
```

Explicación:
Cambia el estado de visibilidad con la propiedad `Visible`.

Ejercicio 115: Usar ProgressBar con Timer

Enunciado:
Aumenta el valor de una `ProgressBar` con un `Timer`.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value < ProgressBar1.Maximum Then
        ProgressBar1.Value += 1
    Else
        Timer1.Stop()
    End If
End Sub
```

Explicación:
El `Timer` aumenta el valor de la barra hasta su máximo.

Ejercicio 116: Cambiar texto de un botón al hacer clic

Enunciado:
El botón cambia su texto cuando se pulsa.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Button1.Text = "¡Gracias!"
End Sub
```

Explicación:
Cambia el valor de la propiedad `Text` del botón.

Ejercicio 117: Calcular el factorial de un número

Enunciado:
Ingresa un número y muestra su factorial.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim n As Integer = Val(TextBox1.Text)
    Dim f As Long = 1
    For i As Integer = 1 To n
        f *= i
    Next
    Label1.Text = "Factorial: " & f
End Sub
```

Explicación:
El bucle calcula el factorial multiplicando desde 1 hasta `n`.

Ejercicio 118: Calcular cuántas vocales tiene un texto

Enunciado:
Cuenta la cantidad de vocales en el contenido de un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim texto As String = TextBox1.Text.ToLower()
    Dim contador As Integer = 0
    For Each c As Char In texto
        If "aeiou".Contains(c) Then contador += 1
    Next
    Label1.Text = "Vocales: " & contador
End Sub
```

Explicación:
Se recorre el texto y se cuenta cada vocal encontrada.

Ejercicio 119: Cambiar el tamaño de fuente de un Label

Enunciado:
Aumenta el tamaño de letra de un `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Font = New Font(Label1.Font.FontFamily, 20)
```

```
End Sub
```

Explicación:
Se establece una nueva fuente con el mismo tipo y mayor tamaño.

Ejercicio 120: Crear reloj digital en tiempo real

Enunciado:
Muestra la hora en tiempo real en un `Label` usando un `Timer`.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = DateTime.Now.ToString("HH:mm:ss")
End Sub
```

Explicación:
Cada tick del `Timer` actualiza la hora con formato de 24 horas.

Ejercicio 121: Aceptar solo letras en un TextBox

Enunciado:
Evita que el usuario escriba números en un `TextBox`.

Solución:

```
Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox1.KeyPress
    If Not Char.IsLetter(e.KeyChar) AndAlso Not Char.IsControl(e.KeyChar) Then
        e.Handled = True
    End If
End Sub
```

Explicación:
Bloquea cualquier carácter que no sea letra o tecla de control.

Ejercicio 122: Cambiar imagen de fondo del formulario

Enunciado:
Permite cargar una imagen como fondo del formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim open As New OpenFileDialog()
    If open.ShowDialog() = DialogResult.OK Then
        Me.BackgroundImage = Image.FromFile(open.FileName)
        Me.BackgroundImageLayout = ImageLayout.Stretch
    End If
End Sub
```

Explicación:
`BackgroundImage` cambia el fondo y `Stretch` lo adapta al tamaño.

Ejercicio 123: Mostrar una lista desplegable con nombres

Enunciado:
Muestra una lista de nombres en un `ComboBox` al cargar el formulario.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ComboBox1.Items.AddRange(New String() {"Ana", "Luis", "Pedro", "Sara"})
End Sub
```

Explicación:
`AddRange` permite agregar varios elementos al `ComboBox`.

Ejercicio 124: Cambiar color de texto en un Label

Enunciado:
Cambia el color de la letra de un `Label` al hacer clic.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.ForeColor = Color.Green
End Sub
```

Explicación:
`ForeColor` define el color del texto.

Ejercicio 125: Seleccionar elemento predeterminado de ComboBox

Enunciado:
Selecciona por defecto el primer elemento del `ComboBox`.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ComboBox1.SelectedIndex = 0
End Sub
```

Explicación:
`SelectedIndex = 0` selecciona el primer ítem automáticamente.

Ejercicio 126: Agregar etiquetas dinámicamente al formulario

Enunciado:
Crea y muestra una etiqueta con texto personalizado desde código.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim lbl As New Label()
    lbl.Text = "Hola desde código"
    lbl.Location = New Point(50, 100)
    Me.Controls.Add(lbl)
End Sub
```

Explicación:
Se crea un nuevo objeto `Label`, se configura y se añade al formulario.

Ejercicio 127: Saludar según el nombre ingresado

Enunciado:
Al pulsar el botón, saluda con el nombre del usuario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Hola, " & TextBox1.Text & "!"
End Sub
```

Explicación:
Concatena el texto con el valor del `TextBox`.

Ejercicio 128: Verificar si un número es par o impar

Enunciado:
Muestra si el número ingresado es par o impar.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim n As Integer = Val(TextBox1.Text)
    If n Mod 2 = 0 Then
        Label1.Text = "Es par"
    Else
        Label1.Text = "Es impar"
    End If
End Sub
```

Explicación:
Se usa el operador `Mod` para verificar si hay resto.

Ejercicio 129: Desplazar un Label horizontalmente con botones

Enunciado:
Dos botones permiten mover un `Label` a la izquierda o derecha.

Solución:

```
Private Sub BtnIzquierda_Click(sender As Object, e As EventArgs) Handles
BtnIzquierda.Click
    Label1.Left -= 10
End Sub

Private Sub BtnDerecha_Click(sender As Object, e As EventArgs) Handles
BtnDerecha.Click
    Label1.Left += 10
End Sub
```

Explicación:
Modifica la propiedad `Left` para mover la posición del `Label`.

Ejercicio 130: Simular cambio de tema claro/oscuro

Enunciado:
Un botón alterna entre fondo claro y oscuro.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If Me.BackColor = Color.White Then
        Me.BackColor = Color.Black
        Label1.ForeColor = Color.White
    Else
        Me.BackColor = Color.White
        Label1.ForeColor = Color.Black
    End If
End Sub
```

Explicación:
Se usa una condición para cambiar los colores del fondo y texto.

Ejercicio 131: Sumar los elementos de un ListBox

Enunciado:
Muestra la suma de todos los elementos numéricos en un `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim suma As Double = 0
    For Each item As String In ListBox1.Items
        suma += Val(item)
    Next
    Label1.Text = "Suma: " & suma
End Sub
```

Explicación:
`Val` convierte cada ítem a número para sumarlo.

Ejercicio 132: Mostrar el mayor de tres números

Enunciado:
Compara tres números ingresados y muestra el mayor.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim a = Val(TextBox1.Text)
    Dim b = Val(TextBox2.Text)
    Dim c = Val(TextBox3.Text)
    Dim mayor = Math.Max(a, Math.Max(b, c))
    Label1.Text = "Mayor: " & mayor
End Sub
```

Explicación:
`Math.Max` devuelve el valor máximo entre dos números.

Ejercicio 133: Alternar color de fondo con CheckBox

Enunciado:
Si se activa el `CheckBox`, el fondo se pone azul. Si se desactiva, blanco.

Solución:

```
Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles
CheckBox1.CheckedChanged
    Me.BackColor = If(CheckBox1.Checked, Color.LightBlue, Color.White)
End Sub
```

Explicación:
`If` condicional asigna un color según el estado del `CheckBox`.

Ejercicio 134: Calcular porcentaje

Enunciado:
Calcula el porcentaje de un valor según lo indicado por el usuario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim valor = Val(TextBox1.Text)
    Dim porcentaje = Val(TextBox2.Text)
    Dim resultado = valor * porcentaje / 100
    Label1.Text = "Resultado: " & resultado
End Sub
```

Explicación:
Se aplica la fórmula de porcentaje básico.

Ejercicio 135: Mostrar números pares del 1 al 100 en ListBox

Enunciado:
Llena un `ListBox` con números pares del 1 al 100.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    ListBox1.Items.Clear()
    For i = 2 To 100 Step 2
        ListBox1.Items.Add(i)
    Next
End Sub
```

Explicación:
Usamos `Step 2` para avanzar de 2 en 2.

Ejercicio 136: Mostrar días de la semana en ComboBox

Enunciado:
Llena un `ComboBox` con los días de la semana.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ComboBox1.Items.AddRange(New String() {"Lunes", "Martes", "Miércoles",
"Jueves", "Viernes", "Sábado", "Domingo"})
End Sub
```

Explicación:
Se utiliza `AddRange` para agregar todos los días a la vez.

Ejercicio 137: Mostrar tabla de multiplicar

Enunciado:
Muestra la tabla de multiplicar del número ingresado del 1 al 10.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim n = Val(TextBox1.Text)
    ListBox1.Items.Clear()
    For i = 1 To 10
        ListBox1.Items.Add(n & " x " & i & " = " & (n * i))
    Next
End Sub
```

Explicación:
Usamos un bucle para generar la tabla y mostrarla.

Ejercicio 138: Mostrar fecha actual

Enunciado:
Muestra la fecha actual al hacer clic en un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = Date.Today.ToString("dd/MM/yyyy")
End Sub
```

Explicación:
`Date.Today` da la fecha actual sin hora.

Ejercicio 139: Cambiar imagen del botón

Enunciado:
Muestra una imagen en un `Button`.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Button1.Image = Image.FromFile("C:\ruta\a\imagen.png")
    Button1.ImageAlign = ContentAlignment.MiddleLeft
End Sub
```

Explicación:
Se asigna una imagen a la propiedad `Image`.

Ejercicio 140: Mostrar día de la semana actual

Enunciado:
Muestra el nombre del día actual (lunes, martes, etc.).

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = Date.Today.ToString("dddd")
End Sub
```

Explicación:
El formato `"dddd"` devuelve el nombre del día.

Ejercicio 141: Alinear texto del Label al centro

Enunciado:
Cambia la alineación del texto del `Label` al centro.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Label1.TextAlign = ContentAlignment.MiddleCenter
End Sub
```

Explicación:
`TextAlign` posiciona el texto dentro del `Label`.

Ejercicio 142: Invertir texto de un TextBox

Enunciado:
Invierte el texto escrito en el `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim texto = TextBox1.Text
    Dim invertido = StrReverse(texto)
    Label1.Text = invertido
End Sub
```

Explicación:
`StrReverse` invierte una cadena.

Ejercicio 143: Verificar si una palabra es palíndromo

Enunciado:
Verifica si el texto ingresado es un palíndromo (igual al revés).

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim texto = TextBox1.Text.ToLower().Replace(" ", "")
    If texto = StrReverse(texto) Then
        Label1.Text = "Es palíndromo"
    Else
        Label1.Text = "No lo es"
    End If
End Sub
```

Explicación:
Se compara el texto con su versión invertida, ignorando espacios.

Ejercicio 144: Salir del programa

Enunciado:
Cerrar la aplicación desde un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Application.Exit()
End Sub
```

Explicación:
`Application.Exit` cierra la aplicación.

Ejercicio 145: Mostrar cantidad de palabras

Enunciado:
Cuenta cuántas palabras contiene un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim palabras = TextBox1.Text.Split(New Char() {" "c},
StringSplitOptions.RemoveEmptyEntries)
    Label1.Text = "Palabras: " & palabras.Length
End Sub
```

Explicación:
Divide el texto por espacios y cuenta los elementos.

Ejercicio 146: Mostrar edad según fecha de nacimiento

Enunciado:
Usa un `DateTimePicker` para calcular la edad del usuario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim edad = Date.Today.Year - DateTimePicker1.Value.Year
    If DateTime.Today < DateTimePicker1.Value.AddYears(edad) Then
        edad -= 1
    End If
    Label1.Text = "Edad: " & edad
End Sub
```

Explicación:
Resta los años y ajusta si aún no ha cumplido años este año.

Ejercicio 147: Contador de clics

Enunciado:
Cuenta cuántas veces se ha hecho clic en un botón.

Solución:

```
Dim contador As Integer = 0

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    contador += 1
    Label1.Text = "Clics: " & contador
End Sub
```

Explicación:
Se incrementa una variable cada vez que se hace clic.

Ejercicio 148: Validar campos obligatorios

Enunciado:
Muestra un mensaje si un `TextBox` está vacío.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If TextBox1.Text.Trim() = "" Then
        MessageBox.Show("El campo no puede estar vacío.")
    End If
End Sub
```

Explicación:
`Trim` elimina espacios, y se comprueba si queda vacío.

Ejercicio 149: Convertir texto a mayúsculas

Enunciado:
Convierte el texto del `TextBox` a mayúsculas.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    TextBox1.Text = TextBox1.Text.ToUpper()
End Sub
```

Explicación:
`ToUpper` convierte la cadena a mayúsculas.

Ejercicio 150: Agregar fecha y hora a ListBox

Enunciado:
Agrega la fecha y hora actual como nueva entrada en el `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    ListBox1.Items.Add(Now.ToString("dd/MM/yyyy HH:mm:ss"))
End Sub
```

Explicación:
`Now.ToString` genera una marca de tiempo legible.

Ejercicio 151: Deshabilitar botón tras primer clic

Enunciado:
Haz que el botón se desactive después de hacer clic una vez.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Button1.Enabled = False
    Label1.Text = "Botón deshabilitado"
End Sub
```

Explicación:
La propiedad `Enabled` del botón se pone en `False` tras hacer clic.

Ejercicio 152: Mostrar nombre seleccionado del ComboBox

Enunciado:
Muestra el nombre seleccionado del `ComboBox` en un `Label`.

Solución:

```
Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ComboBox1.SelectedIndexChanged
    Label1.Text = "Seleccionaste: " & ComboBox1.SelectedItem.ToString()
End Sub
```

Explicación:
El evento `SelectedIndexChanged` se activa al cambiar la selección.

Ejercicio 153: Limpiar todos los TextBox del formulario

Enunciado:
Al hacer clic en un botón, borra el contenido de todos los `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    For Each ctrl As Control In Me.Controls
        If TypeOf ctrl Is TextBox Then
            CType(ctrl, TextBox).Clear()
        End If
    Next
End Sub
```

Explicación:
Recorre todos los controles, y limpia los que sean `TextBox`.

Ejercicio 154: Convertir texto a minúsculas

Enunciado:
Convierte el contenido del `TextBox` a minúsculas.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    TextBox1.Text = TextBox1.Text.ToLower()
End Sub
```

Explicación:
`ToLower()` transforma el texto en minúsculas.

Ejercicio 155: Contar letras en un TextBox

Enunciado:
Muestra cuántas letras tiene el texto escrito.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim letras = TextBox1.Text.Count(Function(c) Char.IsLetter(c))
    Label1.Text = "Letras: " & letras
End Sub
```

Explicación:
Cuenta solo los caracteres alfabéticos.

Ejercicio 156: Activar botón solo si hay texto

Enunciado:
Activa el botón solo si el `TextBox` no está vacío.

Solución:

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles
TextBox1.TextChanged
    Button1.Enabled = Not String.IsNullOrWhiteSpace(TextBox1.Text)
```

```
End Sub
```

Explicación:
Se verifica si el campo contiene texto útil.

Ejercicio 157: Mostrar el número de elementos en un ListBox

Enunciado:
Cuenta cuántos elementos tiene el `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Elementos: " & ListBox1.Items.Count
End Sub
```

Explicación:
`Items.Count` da el total de entradas del `ListBox`.

Ejercicio 158: Quitar último elemento del ListBox

Enunciado:
Elimina el último ítem del `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If ListBox1.Items.Count > 0 Then
        ListBox1.Items.RemoveAt(ListBox1.Items.Count - 1)
    End If
End Sub
```

Explicación:
Se borra el elemento usando su índice.

Ejercicio 159: Detectar tecla Enter en TextBox

Enunciado:
Al presionar Enter, muestra el contenido del `TextBox` en el `Label`.

Solución:

```
Private Sub TextBox1_KeyDown(sender As Object, e As KeyEventArgs) Handles
TextBox1.KeyDown
    If e.KeyCode = Keys.Enter Then
        Label1.Text = TextBox1.Text
    End If
End Sub
```

Explicación:
Se captura el evento `KeyDown` para actuar al presionar Enter.

Ejercicio 160: Mostrar hora actual en tiempo real

Enunciado:
Usa un `Timer` para actualizar la hora en un `Label` cada segundo.

Solución:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = Now.ToString("HH:mm:ss")
End Sub

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Timer1.Interval = 1000
    Timer1.Start()
End Sub
```

Explicación:
El `Timer` actualiza el `Label` cada segundo.

Ejercicio 161: Mostrar mensaje de bienvenida personalizado

Enunciado:
Pide el nombre en un `TextBox` y muestra “Bienvenido, [nombre]”.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = "Bienvenido, " & TextBox1.Text
End Sub
```

Explicación:
Concatenación de texto con el contenido del `TextBox`.

Ejercicio 162: Marcar todas las CheckBox de un grupo

Enunciado:
Marca todas las `CheckBox` del `GroupBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    For Each ctrl In GroupBox1.Controls
        If TypeOf ctrl Is CheckBox Then
            CType(ctrl, CheckBox).Checked = True
        End If
    Next
End Sub
```

Explicación:
Recorre los controles y marca solo los que sean `CheckBox`.

Ejercicio 163: Borrar elemento seleccionado del ListBox

Enunciado:
Elimina el ítem seleccionado en el `ListBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If ListBox1.SelectedIndex >= 0 Then
        ListBox1.Items.RemoveAt(ListBox1.SelectedIndex)
    End If
End Sub
```

Explicación:
Se comprueba si hay selección antes de borrar.

Ejercicio 164: Cambiar tamaño de fuente del Label

Enunciado:
Al hacer clic en el botón, cambia el tamaño de la fuente del `Label`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Font = New Font(Label1.Font.FontFamily, 20)
End Sub
```

Explicación:
Se cambia el tamaño de la fuente manteniendo el tipo original.

Ejercicio 165: Ordenar alfabéticamente un ListBox

Enunciado:
Ordena los elementos del `ListBox` alfabéticamente.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim items = ListBox1.Items.Cast(Of String)().OrderBy(Function(x) x).ToList()
    ListBox1.Items.Clear()
    For Each item In items
        ListBox1.Items.Add(item)
    Next
End Sub
```

Explicación:
Usa LINQ para ordenar y volver a llenar el `ListBox`.

Ejercicio 166: Mostrar selección del RadioButton

Enunciado:
Muestra cuál `RadioButton` fue seleccionado.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If RadioButton1.Checked Then
        Label1.Text = "Seleccionaste opción 1"
    ElseIf RadioButton2.Checked Then
        Label1.Text = "Seleccionaste opción 2"
    End If
End Sub
```

Explicación:
Evalúa cuál botón está marcado.

Ejercicio 167: Mostrar número aleatorio entre 1 y 10

Enunciado:
Genera un número aleatorio entre 1 y 10 y lo muestra.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim rnd As New Random()
    Label1.Text = rnd.Next(1, 11)
End Sub
```

Explicación:
`Next(1, 11)` incluye 1 y excluye 11.

Ejercicio 168: Mostrar mensaje de despedida

Enunciado:
Muestra “Gracias por usar la app” y cierra la aplicación.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    MessageBox.Show("Gracias por usar la app")
    Application.Exit()
End Sub
```

Explicación:
Muestra un mensaje y luego cierra la app.

Ejercicio 169: Seleccionar todo el texto al enfocar TextBox

Enunciado:
Cuando el `TextBox` recibe foco, selecciona todo su texto.

Solución:

```
Private Sub TextBox1_Enter(sender As Object, e As EventArgs) Handles
TextBox1.Enter
    TextBox1.SelectAll()
End Sub
```

Explicación:
`SelectAll()` selecciona todo el texto.

Ejercicio 170: Mostrar caracteres únicos del texto

Enunciado:
Muestra todos los caracteres diferentes del `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim caracteres = TextBox1.Text.Distinct()
    Label1.Text = String.Join(", ", caracteres)
End Sub
```

Explicación:
`Distinct()` filtra letras repetidas.

Ejercicio 171: Convertir texto a código ASCII

Enunciado:
Convierte cada carácter del `TextBox` a su valor ASCII y muéstralo separado por comas.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim resultado = String.Join(", ", TextBox1.Text.Select(Function(c) Asc(c)))
    Label1.Text = resultado
End Sub
```

Explicación:
`Asc(c)` obtiene el valor ASCII de cada carácter.

Ejercicio 172: Aumentar valor de ProgressBar al hacer clic

Enunciado:
Cada clic en el botón debe aumentar el valor del `ProgressBar` en 10.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If ProgressBar1.Value <= 90 Then
        ProgressBar1.Value += 10
    End If
End Sub
```

Explicación:
Se incrementa el valor mientras no supere el máximo (100).

Ejercicio 173: Mostrar cantidad de palabras

Enunciado:
Cuenta y muestra cuántas palabras hay en un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim palabras = TextBox1.Text.Split({" "c},
StringSplitOptions.RemoveEmptyEntries)
    Label1.Text = "Palabras: " & palabras.Length
End Sub
```

Explicación:
Se divide el texto por espacios y se cuentan las partes.

Ejercicio 174: Convertir texto a título (capitalizar)

Enunciado:
Convierte el texto del `TextBox` a formato título: primera letra en mayúscula de cada palabra.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim resultado =
Globalization.CultureInfo.CurrentCulture.TextInfo.ToTitleCase(TextBox1.Text.ToLo
wer())
    TextBox1.Text = resultado
End Sub
```

Explicación:
Utiliza `ToTitleCase` para aplicar formato de título.

Ejercicio 175: Cambiar imagen en PictureBox

Enunciado:
Al hacer clic, cambia la imagen del `PictureBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    PictureBox1.Image = Image.FromFile("C:\imagenes\foto2.jpg")
End Sub
```

Explicación:
Se carga una nueva imagen desde el disco.

Ejercicio 176: Calcular edad a partir de fecha

Enunciado:
Introduce fecha de nacimiento en un `DateTimePicker` y calcula la edad.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim nacimiento = DateTimePicker1.Value
    Dim edad = Date.Now.Year - nacimiento.Year
    If nacimiento > Date.Now.AddYears(-edad) Then edad -= 1
    Label1.Text = "Edad: " & edad
End Sub
```

Explicación:
Se resta el año y se ajusta si no ha cumplido años aún.

Ejercicio 177: Reproducir sonido WAV

Enunciado:
Reproduce un archivo de sonido WAV al hacer clic.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    My.Computer.Audio.Play("C:\sonidos\alerta.wav")
End Sub
```

Explicación:
`My.Computer.Audio.Play` reproduce archivos WAV.

Ejercicio 178: Invertir texto ingresado

Enunciado:
Muestra el texto del `TextBox` al revés.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim invertido = New String(TextBox1.Text.Reverse().ToArray())
    Label1.Text = invertido
End Sub
```

Explicación:
`Reverse()` revierte el orden de los caracteres.

Ejercicio 179: Guardar texto en archivo .txt

Enunciado:
Guarda el contenido de un `TextBox` en un archivo de texto.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    System.IO.File.WriteAllText("C:\archivos\texto.txt", TextBox1.Text)
End Sub
```

Explicación:
`WriteAllText` guarda directamente el texto en un archivo.

Ejercicio 180: Leer archivo .txt y mostrarlo

Enunciado:
Carga el contenido de un archivo `.txt` en un `TextBox`.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If System.IO.File.Exists("C:\archivos\texto.txt") Then
        TextBox1.Text = System.IO.File.ReadAllText("C:\archivos\texto.txt")
    End If
End Sub
```

Explicación:
`ReadAllText` carga todo el archivo como texto.

Ejercicio 181: Dibujar rectángulo en el formulario

Enunciado:
Dibuja un rectángulo rojo al hacer clic en el botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim g As Graphics = Me.CreateGraphics()
```

```
    Dim pen As New Pen(Color.Red, 2)
    g.DrawRectangle(pen, 50, 50, 100, 60)
End Sub
```

Explicación:
`CreateGraphics()` permite dibujar en el formulario.

Ejercicio 182: Cambiar color de fondo del Form

Enunciado:
Cambia el color de fondo del `Form` al presionar un botón.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Me.BackColor = Color.LightBlue
End Sub
```

Explicación:
Modifica la propiedad `BackColor` del formulario.

Ejercicio 183: Validar si un TextBox contiene solo números

Enunciado:
Verifica si el contenido del `TextBox` es numérico.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If IsNumeric(TextBox1.Text) Then
        Label1.Text = "Es un número"
    Else
        Label1.Text = "No es un número"
    End If
End Sub
```

Explicación:
`IsNumeric` valida si es un valor numérico.

Ejercicio 184: Desactivar todos los controles del formulario

Enunciado:
Desactiva todos los controles del formulario.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    For Each ctrl As Control In Me.Controls
        ctrl.Enabled = False
    Next
```

```
End Sub
```

Explicación:
Recorre todos los controles y los desactiva.

Ejercicio 185: Simular una barra de carga

Enunciado:
Usa `Timer` y `ProgressBar` para simular una barra de carga.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ProgressBar1.Minimum = 0
    ProgressBar1.Maximum = 100
    Timer1.Interval = 100
    Timer1.Start()
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If ProgressBar1.Value < 100 Then
        ProgressBar1.Value += 5
    Else
        Timer1.Stop()
        MessageBox.Show("Carga completa")
    End If
End Sub
```

Explicación:
El `Timer` incrementa la barra hasta llenarla.

Ejercicio 186: Cambiar el texto del botón dinámicamente

Enunciado:
Al hacer clic, cambia el texto del botón a “Procesando…”.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Button1.Text = "Procesando..."
End Sub
```

Explicación:
Se actualiza la propiedad `Text` del botón.

Ejercicio 187: Mostrar fecha en formato largo

Enunciado:
Muestra la fecha actual con formato largo.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Text = Date.Today.ToLongDateString()
End Sub
```

Explicación:
`ToLongDateString()` da la fecha completa textual.

Ejercicio 188: Alternar visibilidad de un control

Enunciado:
Oculta o muestra el `Label` al hacer clic.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Label1.Visible = Not Label1.Visible
End Sub
```

Explicación:
Cambia la visibilidad actual del `Label`.

Ejercicio 189: Mostrar número redondeado

Enunciado:
Redondea el número ingresado a dos decimales.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim num As Double = Val(TextBox1.Text)
    Label1.Text = Math.Round(num, 2)
End Sub
```

Explicación:
`Math.Round` limita la cantidad de decimales.

Ejercicio 190: Cambiar idioma del calendario (DateTimePicker)

Enunciado:
Establece el `DateTimePicker` en formato de fecha española.

Solución:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    System.Threading.Thread.CurrentThread.CurrentCulture = New
Globalization.CultureInfo("es-ES")
    DateTimePicker1.Format = DateTimePickerFormat.Long
End Sub
```

Explicación:
Se ajusta la cultura a `es-ES` para fechas en español.

Ejercicio 191: Juego “Adivina el número”

Enunciado:
El programa genera un número aleatorio entre 1 y 100. El usuario introduce un número y el programa indica si es mayor, menor o acertó.

Solución:

```
Dim numeroSecreto As Integer
Dim intentos As Integer = 0

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Dim rnd As New Random()
    numeroSecreto = rnd.Next(1, 101)
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim intento As Integer
    If Integer.TryParse(TextBox1.Text, intento) Then
        intentos += 1
        If intento < numeroSecreto Then
            Label1.Text = "Demasiado bajo"
        ElseIf intento > numeroSecreto Then
            Label1.Text = "Demasiado alto"
        Else
            Label1.Text = "¡Correcto! Intentos: " & intentos
        End If
    Else
        Label1.Text = "Introduce un número válido"
    End If
End Sub
```

Explicación:
Se usa un número aleatorio y se compara con el intento del usuario, dando pistas.

Ejercicio 192: Juego “Piedra, papel o tijeras”

Enunciado:
Usuario selecciona una opción y el programa genera otra al azar, luego decide ganador.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim opciones() As String = {"Piedra", "Papel", "Tijeras"}
    Dim rnd As New Random()
    Dim eleccionPC = opciones(rnd.Next(0, 3))
    Dim eleccionUsuario = ComboBox1.SelectedItem.ToString()

    If eleccionUsuario = eleccionPC Then
        Label1.Text = "Empate! Ambos eligieron " & eleccionPC
    ElseIf (eleccionUsuario = "Piedra" And eleccionPC = "Tijeras") Or
```

```
            (eleccionUsuario = "Papel" And eleccionPC = "Piedra") Or
            (eleccionUsuario = "Tijeras" And eleccionPC = "Papel") Then
        Label1.Text = "Ganaste! PC eligió " & eleccionPC
    Else
        Label1.Text = "Perdiste! PC eligió " & eleccionPC
    End If
End Sub
```

Explicación:
Se compara la elección del usuario y la del PC para determinar el resultado.

Ejercicio 193: Contar clicks rápidos

Enunciado:
El usuario debe hacer el mayor número de clicks en 5 segundos.

Solución:

```
Dim clicks As Integer = 0
Dim tiempo As Integer = 5
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Timer1.Interval = 1000
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    clicks += 1
    Label1.Text = "Clicks: " & clicks
End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
    clicks = 0
    tiempo = 5
    Timer1.Start()
    Label1.Text = "¡Empieza a clickear!"
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    tiempo -= 1
    If tiempo = 0 Then
        Timer1.Stop()
        MessageBox.Show("Tiempo finalizado. Clicks: " & clicks)
    End If
End Sub
```

Explicación:
El botón 2 inicia el juego y un temporizador controla el tiempo para contar clicks.

Ejercicio 194: Juego “Memoria” con botones

Enunciado:
Se muestran 4 botones con colores, el usuario debe recordar el orden y repetirlo.

Solución:

```
Dim secuencia As New List(Of Color)
```

```
Dim intento As New List(Of Color)
Dim rnd As New Random()
Dim index As Integer = 0

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    ' Generar secuencia de 4 colores
    Dim colores() As Color = {Color.Red, Color.Blue, Color.Green, Color.Yellow}
    For i = 1 To 4
        secuencia.Add(colores(rnd.Next(0, colores.Length)))
    Next
    MostrarSecuencia()
End Sub

Private Sub MostrarSecuencia()
    ' Muestra secuencia al usuario (ejemplo simple)
    For Each c In secuencia
        MessageBox.Show("Color: " & c.ToString())
    Next
End Sub

Private Sub BotonColor_Click(sender As Object, e As EventArgs) Handles
Button1.Click, Button2.Click, Button3.Click, Button4.Click
    Dim btn As Button = CType(sender, Button)
    intento.Add(btn.BackColor)

    If intento.Count = secuencia.Count Then
        If intento.SequenceEqual(secuencia) Then
            MessageBox.Show("¡Correcto!")
        Else
            MessageBox.Show("¡Error, inténtalo de nuevo!")
        End If
        intento.Clear()
    End If
End Sub
```

Explicación:
Se genera una secuencia de colores y el usuario debe reproducirla con los botones.

Ejercicio 195: Juego “Snake” simple

Enunciado:
Mover una caja (PictureBox) con las flechas en un panel.

Solución:

```
Private velocidad As Integer = 10

Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles
MyBase.KeyDown
    Select Case e.KeyCode
        Case Keys.Up
            PictureBox1.Top -= velocidad
        Case Keys.Down
            PictureBox1.Top += velocidad
        Case Keys.Left
            PictureBox1.Left -= velocidad
        Case Keys.Right
            PictureBox1.Left += velocidad
    End Select
End Sub
```

Explicación:
Detecta la tecla y mueve el `PictureBox` en la dirección correspondiente.

Ejercicio 196: Juego “Pong” básico

Enunciado:
Mover un rectángulo (PictureBox) con el ratón, y otro con teclado para simular Pong.

Solución:

```
Private Sub Form1_MouseMove(sender As Object, e As MouseEventArgs) Handles
Me.MouseMove
    PictureBox1.Top = e.Y - PictureBox1.Height / 2
End Sub

Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles
MyBase.KeyDown
    If e.KeyCode = Keys.Up Then
        PictureBox2.Top -= 10
    ElseIf e.KeyCode = Keys.Down Then
        PictureBox2.Top += 10
    End If
End Sub
```

Explicación:
Uno se mueve con el ratón, el otro con el teclado.

Ejercicio 197: Juego “Laberinto”

Enunciado:
Mover un `PictureBox` con las flechas dentro de un área, sin salir de límites.

Solución:

```
Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles
MyBase.KeyDown
    Dim paso As Integer = 5
    Select Case e.KeyCode
        Case Keys.Up
            If PictureBox1.Top - paso >= 0 Then PictureBox1.Top -= paso
        Case Keys.Down
            If PictureBox1.Bottom + paso <= Me.ClientSize.Height Then
PictureBox1.Top += paso
        Case Keys.Left
            If PictureBox1.Left - paso >= 0 Then PictureBox1.Left -= paso
        Case Keys.Right
            If PictureBox1.Right + paso <= Me.ClientSize.Width Then
PictureBox1.Left += paso
    End Select
End Sub
```

Explicación:
Controla que el jugador no se salga del área del formulario.

Ejercicio 198: Juego “Dado virtual”

Enunciado:
Al pulsar un botón, muestra un número aleatorio entre 1 y 6.

Solución:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim rnd As New Random()
    Dim dado = rnd.Next(1, 7)
    Label1.Text = "Resultado: " & dado
End Sub
```

Explicación:
Simula la tirada de un dado usando número aleatorio.

Ejercicio 199: Juego “Piedra, papel o tijeras” gráfico

Enunciado:
Usar imágenes para que el usuario elija piedra, papel o tijeras y el PC responde con una imagen.

Solución:

```
Dim opciones() As String = {"piedra", "papel", "tijeras"}
Dim rnd As New Random()
Private Sub PictureBoxUsuario_Click(sender As Object, e As EventArgs) Handles
PictureBoxPiedra.Click, PictureBoxPapel.Click, PictureBoxTijeras.Click
    Dim pic As PictureBox = CType(sender, PictureBox)
    Dim eleccionUsuario = pic.Name.Replace("PictureBox", "").ToLower()
    Dim eleccionPC = opciones(rnd.Next(0, 3))

    PictureBoxPC.Image = Image.FromFile($"C:\imagenes\{eleccionPC}.png")

    ' Lógica similar al ejercicio 192 para decidir ganador
    ' ...
End Sub
```

Explicación:
Cada imagen actúa como botón y se muestra respuesta gráfica.

Ejercicio 200: Juego “Cuenta atrás”

Enunciado:
Temporizador que cuenta hacia atrás desde 10 y muestra un mensaje al acabar.

Solución:

```
Dim tiempo As Integer = 10

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    tiempo = 10
    Timer1.Interval = 1000
    Timer1.Start()
End Sub
```

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If tiempo > 0 Then
        Label1.Text = "Tiempo: " & tiempo
        tiempo -= 1
    Else
        Timer1.Stop()
        MessageBox.Show("¡Tiempo finalizado!")
    End If
End Sub
```

Explicación:
El `Timer` resta segundos y muestra mensaje cuando termina.

Epílogo

Al llegar al final de este libro de ejercicios de Visual Basic .NET, espero que hayas encontrado en estas páginas no solo retos y soluciones, sino también una oportunidad para crecer como programador y consolidar tus habilidades.

La programación es un arte en constante evolución, y Visual Basic .NET es una herramienta poderosa que, bien dominada, te permitirá crear aplicaciones robustas, intuitivas y útiles. Más allá de aprender la sintaxis o entender las estructuras básicas, este libro buscó llevarte paso a paso hacia un entendimiento práctico y sólido mediante ejercicios progresivos y explicaciones claras.

Recuerda que el aprendizaje no termina aquí. Cada línea de código que escribas, cada problema que resuelvas y cada proyecto que emprendas contribuirán a tu experiencia y confianza. No temas explorar más allá de estos ejercicios: investiga nuevas funcionalidades, experimenta con tus propias ideas y adapta lo aprendido a tus necesidades reales.

El camino del desarrollo de software puede ser desafiante, pero también profundamente gratificante. Que este libro sea solo el comienzo de un recorrido apasionante en el mundo de la programación con Visual Basic .NET.

¡Mucho éxito en tus futuros proyectos y que la creatividad y la lógica te acompañen siempre!